

**PENGEMBANGAN DAN ANALISIS APLIKASI SIMULASI TES
BERBASIS *MOBILE APPLICATION* MENGGUNAKAN BAHASA
PEMROGRAMAN *JAVA MICRO EDITION*, *PHP* DAN *MySQL***

TUGAS AKHIR SKRIPSI

Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Sarjana Pendidikan



Disusun Oleh :

Rizam Yudinar

NIM. 08502244021

**PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2015**

HALAMAN PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

**PENGEMBANGAN DAN ANALISIS APLIKASI SIMULASI TES BERBASIS
MOBILE APPLICATION MENGGUNAKAN BAHASA PEMROGRAMAN
JAVA MICRO EDITION, PHP DAN MySQL**

Disusun oleh :

Rizam Yudinar
NIM. 08502244021

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan


Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan

Yogyakarta, 26 Mei 2015

Mengetahui,
Ketua Program Studi
Pendidikan Teknik Elektronika


Muhammad Munir, M.Pd
NIP.19630512 198901 1 001

Disetujui,
Dosen Pembimbing


Totok Sukaridiyono, M.T
NIP.19670930 199303 1 005

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Rizam Yudinar
NIM : 08502244021
Program Studi : Pendidikan Teknik Elektronika
Angkatan : 2008
Judul TAs : Pengembangan Dan Analisis Aplikasi Simulasi Tes
Berbasis *Mobile Application* Menggunakan Bahasa
Pemrograman *Java Micro Editon, PHP Dan MySQL*

Menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim. Demikian pernyataan ini dibuat dalam keadaan sadar dan tidak dipaksakan untuk digunakan sebagaimana mestinya.

Yogyakarta, 25 Mei 2015

Penulis



Rizam Yudinar

NIM. 08502244021


HALAMAN PENGESAHAN

Tugas Akhir Skripsi

PENGEMBANGAN DAN ANALISIS APLIKASI SIMULASI TES BERBASIS MOBILE APPLICATION MENGGUNAKAN BAHASA PEMROGRAMAN JAVA MICRO EDITION, PHP DAN MySQL

Disusun oleh:
Rizam Yudinar
NIM. 08502244021

Telah dipertahankan di depan Tim Penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta
Pada tanggal 28 Agustus 2015

Tim Penguji		
Nama/Jabatan	Tanda Tangan	Tanggal
Totok Sukardiyono, M.T Ketua Penguji/Pembimbing		2-9/2015
Ponco Wali Pranoto, M.Pd Sekretaris		2-9/2015
Achmad Fatchi, M.Pd Penguji Utama		2-9/2015

Yogyakarta, 28 Agustus 2015

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,


Dr. Moch. Bruri Triyono
NIP. 19560216 198603 1 003

MOTTO DAN PERSEMBAHAN

MOTTO

“Dan bahwasanya seorang manusia tiada memperoleh selain apa yang telah diusahakannya” - Q.S. An Najm : 39.

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan ada kemudahan.” – Q.S. Al Insyirah : 5-6

“Sesungguhnya Allah tidak akan mengubah nasib suatu kaum hingga mereka mengubah diri mereka sendiri.” - Q.S. Ar-Ra’d : 11

“Allah tidak membebani seseorang itu melainkan sesuai dengan kesanggupannya.” -Q.S. Al-Baqarah : 286

“Man jadda wa jadda, man shabara zhafira” – siapa yang sungguh-sungguh akan berhasil, siapa yang bersabar akan beruntung

“Ilmu tanpa agama adalah lumpuh, agama tanpa ilmu adalah buta.”- Albert Einstein

PERSEMBAHAN

Karya kecil ini kupersembahkan untuk:

Allah SWT, penciptaku yang selalu mengetahui segala keluh kesahku, hanya kepada-Mu lah hamba menyembah dan memohon.

Rasuulullaah Muhammad SAW panutan dalam hidupku.

Ayahanda Ichwan Abror, Ibunda Fatonah Nurhidayati yang selalu memberikan dukungan, do’a, pengertian, motivasi, dan kasih sayang yang tiada henti demi terselesainya karya ini.

Saudara tercinta Azmi Gading Lazuardi dan Ivan Langit Suryaning Bumi yang selalu memberikan dukungan, do’a, pengertian, motivasi, dan kasih sayang yang tiada henti demi terselesainya karya ini.

Keluarga H. M. Muchlas Abror, yang selalu memberikan dukungan, do’a, pengertian, motivasi, dan kasih sayang yang tiada henti demi terselesaikannya karya ini.

Anjar Fiandrianto, Bonita Destiana, Novita Anindita, Agus Irawan, Banu Desi Antoro, Ahmad Faiq Abror, Adhi Wicaksono, Yanuar Arifin, Refany Anhar, Zumrotul Ahkamiyati, Lisa Noor Ardhiani, Fahmawati Fajrin atas segala do'a, keceriaan, dukungan, bantuan dan semangatnya.

Bayu Aryoyudanta, S.Pd, Lukman Rian Affandi, M.Pd, Arwan Nur Ramadhan, M.Pd atas segala bantuannya.

Keluarga UKMF Matriks, yang selalu memberikan keceriaan, do'a, dan semangat.

Keluarga Dinamid Research Grup yang selalu memberikan keceriaan, do'a, dan semangat.

Keluarga Halan-Halan Hore yang selalu memberikan keceriaan, do'a, dan semangat.

Teman-teman seperjuangan Pendidikan Teknik Elektronika kelas D angkatan 2008.

**PENGEMBANGAN DAN ANALISIS APLIKASI SIMULASI TES BERBASIS
MOBILE APPLICATION MENGGUNAKAN BAHASA PEMROGRAMAN
JAVA MICRO EDITION, PHP DAN MySQL**

Oleh :

Rizam Yudinar
NIM. 08502244021

ABSTRAK

Tujuan penelitian ini untuk: (1) mengembangkan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*, (2) mengetahui hasil pengujian kualitas aplikasi Simulasi Tes berdasarkan *McCall's Software Quality* yang meliputi aspek *correctness, reliability, efficiency, integrity, usability, maintainability, flexibility, testability, reusability, interoperability*.

Metode penelitian yang digunakan adalah *Research and Development* (R&D). Tahap-tahap dalam penelitian ini meliputi: (1) analisis (analisis kebutuhan, analisis materi, analisis *software*), (2) desain (desain sistem dan desain *interface*), (3) pengembangan (perancangan desain *interface* dan pembuatan program), (4) pengujian (uji kualitas berdasarkan *McCall's Software Quality*). Sumber data/subjek penelitian yaitu 35 siswa SMK Negeri 3 Yogyakarta kelas XI jurusan Audio Video untuk pengujian *usability*, 5 orang ahli untuk pengujian *reliability, efficiency, integrity, maintainability, flexibility, testability, reusability, interoperability* dan dokumentasi kode sumber program untuk pengujian *correctness*. Teknik dalam pengumpulan data adalah observasi, studi literatur, serta instrumen penelitian.

Hasil penelitian diketahui bahwa: (1) Pengembangan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL* dilakukan menggunakan *software* NetBeans IDE 7.0.1 dengan kategori 2 pengguna yaitu: admin/guru (*login*, pilih soal, kerjakan soal, *upload* soal, login aplikasi web, melihat nilai, pembahasan, pengayaan, *entry* data user/siswa, *entry* database materi, *logout*) dan siswa (*login*, pilih soal, kerjakan soal, *upload* soal, login aplikasi web, melihat nilai, pembahasan, pengayaan, *logout*), (2) hasil pengujian kualitas aplikasi Simulasi Tes berbasis *mobile application* diperoleh nilai *correctness* sebesar 11.8, 7.58 dan 8.29 yang berada pada rentang 0-40 *error per KLOC*, *reliability* sebesar 91.43 %, *efficiency* sebesar 92.50 %, *integrity* sebesar 93.33 %, *maintainability* sebesar 94.29 %, *flexibility* sebesar 92.50 %, *testability* sebesar 89.33 %, *reusability* sebesar 90.00 %, *interoperability* sebesar 80.00 % dan *usability* menghasilkan nilai alpha cronbach 0.965 (Sangat Baik).

Kata kunci: simulasi tes, mobile application, McCall's software quality

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka untuk memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan judul “Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*” dapat disusun sesuai dengan harapan. Tugas Akhir Skripsi ini dapat diselesaikan tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkenaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Totok Sukardiyono M.T. selaku pembimbing penelitian yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini.
2. Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, Ph.D, Rahmatul Irfan, M.T, Dra. Rr. Sri Handayani, Agus Manaji, S.Si, Dra. Veronika Rita Rini W. Soesapto Joeni Hantoro, S.T, Bayu Aryoyudanta, S.Pd, Bemado Rizky Julhenry, Anjar Fiandrianto, Yanuar Arifin, Refany Anhar, S.Pd selaku validator dalam penelitian TAS yang memberikan saran/masukan perbaikan sehingga penelitian TAS dapat terlaksana sesuai tujuan.
3. Muhammad Munir, M.Pd, selaku Ketua Jurusan Pendidikan Teknik Elektronika yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesainya Tugas Akhir Skripsi ini.
4. Dr. Moch. Bruri Triyono selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi.

5. Drs. Aruji Siswanto selaku kepala sekolah SMK Negeri 3 Yogyakarta yang telah memberi izin dan bantuan dalam pelaksanaan Tugas Akhir Skripsi ini.
6. Para guru dan staf SMK Negeri 3 Yogyakarta yang telah memberi bantuan memperlancar pengambilan data selama proses penelitian Tugas Akhir Skripsi ini.
7. Direktur PT. Nuwira dan Direktur PT. Sebangsa Bersama atas izin kesempatan, bantuan, serta kerjasamanya yang baik sehingga penelitian dapat berjalan dengan lancar.
8. Semua pihak, secara langsung maupun tidak langsung yang tidak dapat disebutkan di sini atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini.

Penulis mengucapkan banyak terima kasih, semoga Allah SWT selalu melimpahkan karunia, hidayah dan ilmu yang bermanfaat bagi kita semua. Penulis berharap semoga Tugas Akhir Skripsi ini dapat bermanfaat bagi penulis dan para pembaca. Amin.

Yogyakarta, 25 Mei 2015
Penulis



Rizam Yudinar
NIM. 08502244021

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PERSETUJUAN	ii
SURAT PERNYATAAN	iii
HALAMAN PENGESAHAN.....	iv
MOTTO DAN PERSEMBAHAN.....	iv
ABSTRAK.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xv
DAFTAR GAMBAR	xviii
DAFTAR LAMPIRAN	xxi
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Identifikasi Masalah	5
C. Batasan Masalah	6
D. Rumusan Masalah.....	6
E. Tujuan	7
F. Manfaat.....	7
BAB II KAJIAN PUSTAKA	9
A. KAJIAN TEORI	9
1. Definisi Aplikasi	9

2. Aplikasi Simulasi	10
3. Metode Pembelajaran Simulasi	10
4. Simulasi	12
5. Definisi <i>Software</i>	13
6. Tes	15
7. Tes Pilihan Berganda	21
8. Aplikasi Berbasis <i>Handphone (Mobile Application)</i>	25
9. Pengenalan Java	27
10. Bahasa Pemrograman PHP (<i>Hypertext Preprocessor</i>)	33
11. <i>MySQL</i>	34
12. <i>Software Quality</i>	36
13. UML (<i>Unified Modeling Language</i>)	37
14. Aplikasi <i>Client-Server</i>	40
15. <i>Software Development Life Cycle (SDLC)</i>	43
16. <i>McCall's Quality Factors</i>	44
17. Pengujian <i>Software</i>	55
B. Kajian Penelitian Yang Relevan	63
C. Kerangka Berfikir	65
D. Pertanyaan Peneliti	68
BAB III METODE PENELITIAN	69
A. Jenis Penelitian	69
B. Desain Penelitian	70
1. Analisis (<i>Analysis</i>)	71
2. Desain Sistem (<i>Design</i>)	73

3. Pengembangan	73
4. Pengujian (<i>Evaluation</i>)	74
C. Sumber Data.....	76
1. Subjek Penelitian.....	76
2. Sampel Penelitian	76
3. Variabel Penelitian	77
4. Tempat dan Waktu Penelitian	77
D. Metode dan Alat Pengumpulan Data	78
1. Aspek <i>Correctness</i>	79
2. Aspek <i>Reliability</i>	79
3. Aspek <i>Efficiency</i>	79
4. Aspek <i>Integrity</i>	80
5. Aspek <i>Usability</i>	80
6. Aspek <i>Maintainability</i>	81
7. Aspek <i>Flexibility</i>	81
8. Aspek <i>Testability</i>	81
9. Aspek <i>Reusability</i>	82
10. Aspek <i>Interoperability</i>	82
E. Validitas dan Reliabilitas Instrumen	82
1. Validitas	82
2. Reliabilitas	88
F. Teknik Analisis Data	89
1. Analisis <i>White Box Testing</i>	89
2. Analisis <i>Black Box Testing</i>	90
3. Analisis <i>Alpha Testing</i>	90

4. Analisis <i>Beta Testing</i>	92
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	93
A. Pengembangan Perangkat Lunak	93
1. Analisis Kebutuhan Sistem	93
2. Analisis Kebutuhan Fitur	93
3. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak	95
B. Tahapan Desain	96
4. Desain Sistem	96
5. Perancangan <i>Unified Modeling Language</i>	96
6. Desain <i>Database</i>	124
C. <i>Development</i> /Pengembangan.....	127
1. Implementasi Isi Soal Materi Menggunakan PHP dan MySQL...128	
2. Implementasi Logika Program Menggunakan Bahasa Pemrograman Java.....	129
3. Implementasi <i>User Interface</i>	129
4. Implementasi Sistem User	136
D. Pengujian Perangkat Lunak	141
1. <i>White Box Testing</i>	141
2. <i>Black Box Testing</i>	143
E. <i>Alpha Testing</i>	146
F. <i>Beta Testing</i>	150
1. Pengujian Aspek <i>Usability</i>	150
BAB V SIMPULAN DAN SARAN	153
A. Simpulan.....	153

B. Saran	156
DAFTAR PUSTAKA	158
LAMPIRAN	162

DAFTAR TABEL

	Halaman
Tabel 1. <i>Platform Java</i> (Taptozani, 2009: 10)	27
Tabel 2. Metode Perhitungan <i>Correctness</i> (Pressman, 2010: 709)	56
Tabel 3. Ukuran Proyek dan Densitas <i>Error</i> (McConnel, 2004: 698)	56
Tabel 4. Sub-karakteristik Aspek <i>Reliability</i> (Pressman, 2000)	56
Tabel 5. Sub-karakteristik Aspek <i>Efficiency</i> (Pressman, 2000)	57
Tabel 6. Sub-karakteristik Aspek <i>Integrity</i> (Pressman, 2000)	58
Tabel 7. Sub-karakteristik Aspek <i>Usability</i> (Pressman, 2000)	59
Tabel 8. Sub-karakteristik Aspek <i>Maintanability</i> (Pressman, 2000)	60
Tabel 9. Sub-karakteristik Aspek <i>Flexibility</i> (Pressman, 2000)	60
Tabel 10. Sub-karakteristik Aspek <i>Testability</i> (Pressman, 2000)	61
Tabel 11. Sub-karakteristik Aspek <i>Reusability</i> (Pressman, 2000)	62
Tabel 12. Sub-karakteristik Aspek <i>Interoperability</i> (Pressman, 2000)	63
Tabel 13. Spesifikasi Hardware yang Digunakan	72
Tabel 14. <i>Computer System Usability Questionnaire</i> oleh J.R. Lewis (Lewis, 1993)	80
Tabel 15. Hasil Perhitungan <i>CVR</i>	84
Tabel 16. Validitas Instrumen <i>Usability</i>	85
Tabel 17. Hasil Perhitungan Validitas Instrumen <i>Reliability</i>	85
Tabel 18. Hasil Perhitungan Validitas Instrumen <i>Efficiency</i>	86
Tabel 19. Hasil Perhitungan Validitas Instrumen <i>Integrity</i>	86
Tabel 20. Hasil Perhitungan Validitas Instrumen <i>Maintanability</i>	86
Tabel 21. Hasil Perhitungan Validitas Instrumen <i>Flexibility</i>	87

Tabel 22. Hasil Perhitungan Validitas Instrumen <i>Testability</i>	87
Tabel 23. Hasil Perhitungan Validitas Instrumen <i>Reusability</i>	87
Tabel 24. Hasil Perhitungan Validitas Instrumen <i>Interoperability</i>	88
Tabel 25. Reliabilitas Instrumen	88
Tabel 26. Ukuran Proyek dan Densitas <i>Error</i> (Pressman, 2010: 709)	90
Tabel 27. Interpretasi Persentase Likert.....	91
Tabel 28. Nilai Konsistensi Internal Alpha Cronbach	92
Tabel 29. <i>Use Case</i> Proses Memulai Aplikasi	99
Tabel 30. <i>Use Case</i> Proses <i>Login</i>	100
Tabel 31. <i>Use Case</i> Menu Pilih Soal.....	100
Tabel 32. <i>Use Case</i> Menu <i>Input</i> Jawaban.....	100
Tabel 33. <i>Use Case</i> Proses <i>Upload</i> Nilai.....	101
Tabel 34. <i>Use Case</i> <i>Login</i> Aplikasi Web	101
Tabel 35. <i>Use Case</i> Melihat Nilai Aplikasi Web.....	102
Tabel 36. <i>Use Case</i> Menu Pembahasan Aplikasi Web	102
Tabel 37. <i>Use Case</i> Menu Pengayaan Pada Aplikasi Web.....	102
Tabel 38. Struktur Tabel Pada <i>Database</i>	124
Tabel 39. Detail Tabel Mapel.....	125
Tabel 40. Detail Tabel Nilai.....	125
Tabel 41. Detail Tabel Pembahasan	125
Tabel 42. Detail Tabel Tes.....	126
Tabel 43. Detail Tabel Tipe.....	126
Tabel 44. Detail Tabel <i>User</i>	126
Tabel 45. Pengujian Aspek <i>Correctness</i>	141
Tabel 46. Metode Perhitungan <i>Correctness</i> (Pressman, 2010:709)	142

Tabel 47. Pengujian Aspek <i>Correctness</i>	142
Tabel 48. Uji <i>Black-box Testing</i> Menjalankan Aplikasi.....	143
Tabel 49. Uji <i>Black-box Login</i> Aplikasi	144
Tabel 50. Uji <i>Black-box Login</i> Aplikasi	144
Tabel 51. Uji <i>Black-box Menu Test</i>	144
Tabel 52. Uji <i>Black-box Menu Test</i>	145
Tabel 53. Uji <i>Black-box Menu Login</i> Aplikasi Web.....	145
Tabel 54. Uji <i>Black-box Menu Nilai</i>	145
Tabel 55. Uji <i>Black-box Menu Pembahasan Soal</i>	146
Tabel 56. Uji <i>Black-box Menu Pengayaan</i>	146
Tabel 57. Uji <i>Black-box Menu Logout</i>	146
Tabel 58. Hasil Perhitungan <i>Alpha Testing</i>	148
Tabel 59. Persentase Kelayakan Tiap Faktor	150
Tabel 60. Hasil Pengujian <i>Usability</i>	151
Tabel 61. Hasil Perhitungan <i>Alpha Cronbach</i>	152

DAFTAR GAMBAR

	Halaman
Gambar 1. Statistik <i>Mobile Operating System</i> di Indonesia Periode Juli 2012 – Juli 2013 (StatCounter Global Stats, 2013).....	4
Gambar 2. <i>Platform</i> Java (Taptozani, 2009: 10)	28
Gambar 3. Arsitektur Java ME (Taptozani, 2009:12).....	29
Gambar 4. Arsitektur Umum <i>Client Server</i> (Taptozani, 2009: 33).....	41
Gambar 5. <i>McCall's Software Quality Factors</i> (Pressman R. S., 2000)	44
Gambar 6. Hubungan McCall's <i>Quality Factor</i> dengan <i>Metric</i> Pengujian (Pressman R. S., 2000)	45
Gambar 7. Kerangka Berpikir	67
Gambar 8. Waterfall Method (Pressman, 2001: 29)	70
Gambar 9. Prosedur dan Alur Pengembangan	71
Gambar 10. Gambaran Umum Aplikasi Simulasi Tes.....	96
Gambar 11. Rancangan <i>Use Case Diagram</i>	99
Gambar 12. Rancangan <i>Class Diagram</i>	103
Gambar 13. <i>Activity Diagram Sequence Login</i>	104
Gambar 14. <i>Activity Diagram</i> “Menu” Pilih Soal	105
Gambar 15. <i>Activity Diagram</i> “Test”	106
Gambar 16. <i>Activity Diagram</i> “Upload” Nilai.....	107
Gambar 17. <i>Activity Diagram Login</i> Aplikasi Web.....	108
Gambar 18. <i>Activity Diagram</i> Menu “Nilai” Aplikasi Web	109
Gambar 19. <i>Activity Diagram</i> Menu Pembahasan	110
Gambar 20. <i>Activity Diagram</i> Menu Pengayaan	111

Gambar 21. <i>Sequence</i> “Login”	112
Gambar 22. <i>Sequence</i> “Menu” Pilihan Soal.....	112
Gambar 23. <i>Sequence</i> Menu “Test”	113
Gambar 24. <i>Sequence</i> “Upload” Nilai.....	114
Gambar 25. <i>Sequence</i> “Login” Aplikasi Web	115
Gambar 26. <i>Sequence</i> Menu “Nilai”	116
Gambar 27. <i>Sequence</i> Menu Pengayaan	117
Gambar 28. <i>Sequence</i> Menu Pembahasan	118
Gambar 29. Rancangan Tampilan Awal <i>Splash Screen</i>	119
Gambar 30. Rancangan Tampilan Halaman <i>Login</i>	119
Gambar 31. Rancangan Tampilan Halaman Pilih Soal	120
Gambar 32. Rancangan Tampilan Halaman Soal	120
Gambar 33. Rancangan Tampilan <i>Upload</i> Nilai.....	121
Gambar 34. Rancangan Tampilan “ <i>Login</i> ”	121
Gambar 35. Rancangan Tampilan Menu Utama	122
Gambar 36. Rancangan Tampilan Menu Nilai	122
Gambar 37. Rancangan Tampilan Menu “Pengayaan”	123
Gambar 38. Rancangan Tampilan Menu “Pembahasan”	124
Gambar 39. Potongan <i>Script</i> XML.....	127
Gambar 40. Potongan <i>Script</i> PHP.....	128
Gambar 41. Potongan <i>Script</i> Halaman <i>Splash Screen</i>	130
Gambar 42. Implementasi Halaman <i>Splash Screen</i>	130
Gambar 43. Potongan <i>Script</i> Halaman <i>Login</i>	131
Gambar 44. Implementasi Halaman Form Login	132
Gambar 45. Potongan <i>Script</i> Halaman Menu Soal	133

Gambar 46. Implementasi Halaman Menu Soal	133
Gambar 47. Potongan <i>Script</i> Halaman Soal	134
Gambar 48. Implementasi Halaman <i>Input</i> Jawaban	135
Gambar 49. Implementasi Halaman Upload Nilai	135
Gambar 50. Pengujian Java 2.0 Bagian <i>Splash Screen</i>	136
Gambar 51. Pengujian Java 2.0 Bagian Halaman <i>Form Login</i>	136
Gambar 52. Pengujian Java 2.1 Bagian <i>Splash Screen</i>	137
Gambar 53. Pengujian Java 2.1 Bagian Halaman <i>Form Login</i>	137
Gambar 54. Pengujian Java 2.1 Bagian Menu Soal	137
Gambar 55. Pengujian Java 2.1 Bagian Menu Soal	137
Gambar 56. Pengujian Java 2.1 Bagian Halaman <i>Input</i> Jawaban	138
Gambar 57. Pengujian Java 2.1 Bagian Halaman <i>Upload</i> Nilai	138
Gambar 58. Pengujian Java 2.1 Bagian Halaman <i>Upload</i> Nilai	138
Gambar 59. Pengujian Java 2.0 Bagian Menu Soal	139
Gambar 60. Pengujian Java 2.0 Bagian Menu Soal	139
Gambar 61. Pengujian Java 2.0 Bagian Halaman Input Jawaban	139
Gambar 62. Pengujian Java 2.0 Bagian Halaman <i>Upload</i> Nilai	140
Gambar 63. Pengujian Java 2.0 Bagian Halaman <i>Upload</i> Nilai	140

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Materi Soal Bahasa Inggris Aplikasi Simulasi Tes.....	163
Lampiran 2. Materi Soal Fisika Aplikasi Simulasi Tes.....	171
Lampiran 3. Materi Soal Fisika Aplikasi Simulasi Tes.....	180
Lampiran 4. Surat Ijin Penelitian Fakultas	189
Lampiran 5. Surat Ijin Penelitian BAPPEDA	190
Lampiran 6. Surat Pernyataan Validasi	191
Lampiran 7. Surat Pernyataan Validasi	192
Lampiran 8. Pernyataan Validasi Materi Kimia	193
Lampiran 9. Pernyataan Validasi Materi Bahasa Inggris	194
Lampiran 10. Pernyataan Validasi Materi Fisika	195
Lampiran 11. Revisi Kisi-Kisi dan Instrumen <i>Reliability</i>	196
Lampiran 12. Revisi Kisi-Kisi dan Instrumen <i>Efficiency</i>	198
Lampiran 13. Revisi Kisi-Kisi dan Instrumen <i>Integrity</i>	200
Lampiran 14. Revisi Kisi-Kisi dan Instrumen <i>Maintanability</i>	203
Lampiran 15. Revisi Kisi-Kisi dan Instrumen <i>Flexibility</i>	206
Lampiran 16. Revisi Kisi-Kisi dan Instrumen <i>Testability</i>	209
Lampiran 17. Revisi Kisi-Kisi dan Instrumen <i>Reusability</i>	213
Lampiran 18. Revisi Kisi-Kisi dan Instrumen <i>Interoperability</i>	216
Lampiran 19. Hasil Uji <i>Reliability</i>	218
Lampiran 20. Hasil Uji <i>Efficiency</i>	224
Lampiran 21. Hasil Uji <i>Integrity</i>	230
Lampiran 22. Hasil Uji <i>Maintanability</i>	236

Lampiran 23. Hasil Uji <i>Flexibility</i>	242
Lampiran 24. Hasil Uji <i>Testability</i>	248
Lampiran 25. Hasil Uji <i>Reusability</i>	253
Lampiran 26. Hasil Uji <i>Interoperability</i>	259
Lampiran 27. Hasil Uji <i>Usability</i>	265
Lampiran 28. Surat Keterangan Telah Penelitian	268

BAB I

PENDAHULUAN

A. Latar Belakang

Evaluasi merupakan bagian penting dalam proses pembelajaran, antara lain: sebagai gambaran tentang tingkat penguasaan siswa terhadap satu materi, memberikan gambaran tentang kesulitan belajar siswa dan memberi gambaran tentang posisi siswa di antara kawan-kawannya. Evaluasi pembelajaran Ujian Nasional menjadi salah satu contoh penerapan evaluasi pembelajaran yang berskala besar di Negara Indonesia untuk menunjukkan indikasi kemampuan akhir siswa. Perkembangan UN (Ujian Nasional) sampai saat ini menghasilkan banyak pro dan kontra terhadap hasil belajar siswa. Polemik ini muncul disebabkan oleh salah satu mekanisme evaluasi yang kurang maksimal. Adapun beberapa kendala yang terjadi dalam mekanisme seiring dengan periode Ujian Nasional di akhir tahun ajaran, antara lain: (1) Mengharuskan peserta untuk hadir di kelas pada jam-jam tertentu (seringkali terjadi bentrok dengan kegiatan rutin) masih menggunakan kertas ujian dan memungkinkan hasil akan tercecer ataupun terselip dengan dokumen lain, (2) Memerlukan waktu yang cukup untuk penilaian dan memroses pengolahan skor. (3) Memerlukan waktu yang cukup untuk pemberian umpan balik kepada responden. (4) Secara psikologis evaluasi manual sering menimbulkan kecemasan kepada peserta tes (Setemen, 2010).

Mengacu pada permasalahan evaluasi pembelajaran dalam pendidikan di Indonesia saat ini masih banyak menggunakan media atau instrumen manual berbahan dasar kertas. Ditinjau dari kualitas media berbahan dasar kertas memiliki beberapa kelemahan, antara lain: (a) Mudah kotor (b) Mudah sobek (c)

Mudah basah (d) Tidak *real-time*. Instrumen manual tersebut saat ini mulai memberikan dampak yang kurang maksimal bagi perkembangan dunia pendidikan di Indonesia. Beberapa hambatan evaluasi masih menunjukkan kurangnya efektivitas dalam belajar (Wijaya, 2010). Selain itu, penyelenggara dihadapkan dengan persiapan alat tulis sebagai media evaluasi konvensional dan tidak terlepas dari banyaknya waktu maupun biaya yang diperlukan. Ditambah keterbatasan sumber daya dalam mengoreksi hasil ujian akan memungkinkan terjadi kesalahan data hasil yang diperoleh siswa.

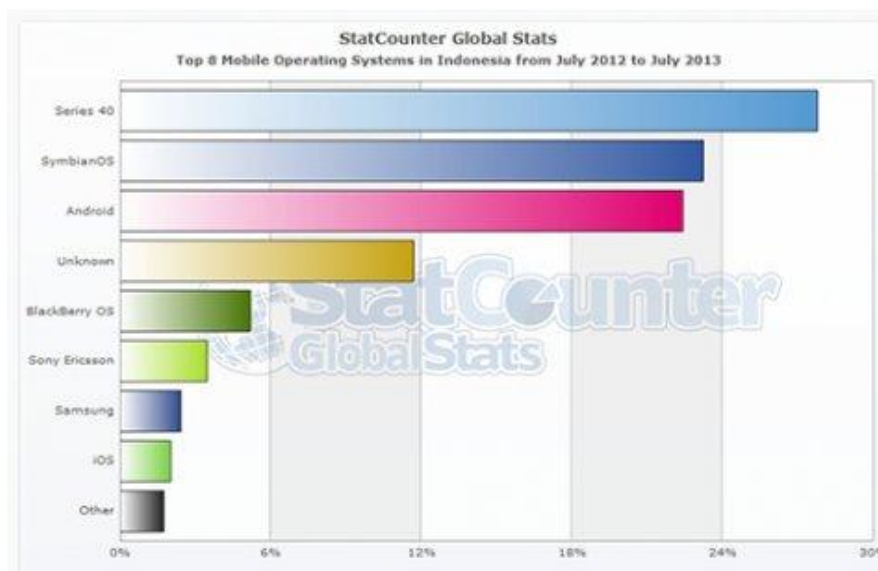
Melihat keterbatasan evaluasi pembelajaran yang ada maka muncul sebuah tuntutan bagi guru untuk mengembangkan suatu model pembelajaran baru. Yaitu, model evaluasi pembelajaran dengan memanfaatkan teknologi informasi dan komunikasi (Hobban & Ferry, 2006). Model evaluasi pembelajaran menggunakan *mobile learning* secara *online* memberikan fleksibilitas dalam memilih waktu dan tempat untuk mengakses soal-soal tes. Dengan syarat para peserta tes terhubung dengan server di mana soal-soal tes disimpan. Peserta dan penyelenggara tes tidak perlu direpotkan lagi dengan penyediaan alat tulis konvensional. Sehingga biaya untuk penyediaan alat tulis dapat dialokasikan untuk biaya lainnya. Hal tersebut merupakan salah satu keuntungan dari pemanfaatan media *online* sebagai sarana untuk mengukur kemampuan belajar serta memacu prestasi seseorang. Berikut adalah beberapa manfaat penggunaan evaluasi menggunakan media secara *online* menurut Iwan Sonjaya, Kuspriyanto & Aciek (2010), antara lain: (a) Arus informasi tetap mengalir setiap waktu tanpa ada batasan waktu dan tempat. (b) Kemudahan mendapatkan *resource* (sumber) yang lengkap. (c) Aktivitas pembelajaran pelajar meningkat. (d) Daya tampung meningkat. (e) Adanya standarisasi pembelajaran. (f)

Meningkatkan *learning outcomes* baik kuantitas/kualitas. Menurut Saepudin (2003), menyatakan bahwa pada jenjang dan jalur pendidikan lain di mana proses belajarnya relatif masih konvensional (tatap muka), yang sesungguhnya sudah tidak lagi mampu memenuhi kebutuhan pendidikan untuk masyarakat yang semakin kompleks, memerlukan inovasi dan media yang mampu menanggulangnya.

Senada dengan hal tersebut, permasalahan ini muncul di SMK Negeri 3 Yogyakarta. Berdasarkan 50 angket yang disebar penulis pada siswa kelas XII AV pada tanggal 22 Januari 2014, menyatakan bahwa 40 di antaranya menyatakan memiliki komputer di rumah dan keseluruhannya memiliki telepon seluler (*handphone*) serta 22 di antaranya memiliki *smartphone*. Namun, perangkat *mobile* ini pada umumnya hanya digunakan untuk SMS (*Short Message Service*), telepon, *chatting*, internet dan hiburan-hiburan seperti permainan dan musik.

Seiring perkembangan teknologi saat ini, semakin banyak kalangan guru dan siswa sudah menggunakan *handphone* sebagai alat komunikasi, untuk mengirim pesan, menerima panggilan. Saat ini, pemanfaatan perangkat bergerak (*mobile device*) dalam proses pembelajaran dikenal sebagai *mobile learning (m-learning)* (Georgiev & Tsvetozar, 2004). O'Malley (2003: 6) mendefinisikan *mobile learning* sebagai suatu pembelajaran bagi siswa (*learner*) untuk tidak diam pada satu tempat atau kegiatan pembelajaran yang terjadi ketika siswa (*learner*) memanfaatkan perangkat teknologi bergerak. Teknologi *mobile learning* telah banyak didominasi oleh ponsel pintar (*smartphone*). Pada April 2010, *Symbian* OS berhasil mendominasi pasar *mobile* serta berhasil dinobatkan sebagai *operating system mobile*. Data tersebut diperkuat dari analisis StatCounter untuk

wilayah Indonesia pada periode bulan Juli 2012 sampai Juli 2013, mayoritas pengguna *feature phone* dikuasai oleh ponsel berbasis *Symbian*. *Symbian OS* masih mendominasi dibandingkan dengan perangkat lain seperti: *Android*, *iOS*, *Windows Phone* dan lainnya seperti ditunjukkan pada gambar 1.



Gambar 1. Statistik *Mobile Operating System* di Indonesia Periode Juli 2012 – Juli 2013 (StatCounter Global Stats, 2013)

Berdasarkan permasalahan yang ada, maka perlu dikembangkan media untuk pengukuran kemampuan siswa yang efektif dan responsif dengan memanfaatkan teknologi ponsel pintar yang sedang berkembang saat ini. Media untuk evaluasi pembelajaran ini dapat digunakan dalam proses pembelajaran maupun di luar proses pembelajaran. Sehingga tidak terbatas penerapannya. Dalam penelitian ini, penulis mengembangkan media “Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*” yang dapat digunakan untuk membantu siswa dalam mengevaluasi kemampuan pembelajaran di tingkat Sekolah Menengah Kejuruan. Aplikasi Simulasi Tes dikembangkan dengan desain yang berbeda dengan aplikasi yang sudah ada. Sehingga belum diketahui kualitas dari perangkat lunak

tersebut. Untuk mengetahui kualitas aplikasi Simulasi Tes Berbasis *Mobile Application* digunakan pengujian kualitas perangkat lunak sesuai standar kelayakan berdasarkan McCall's *Software Quality Factor*. Pengujian kualitas Aplikasi Simulasi Tes Berbasis *Mobile Application* meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintainability*, *flexibility*, *testability*, *reusability* dan *interoperability*. Penelitian yang berjudul "Pengembangan Dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*" bertempat di SMK Negeri 3 Yogyakarta. Melalui perangkat lunak aplikasi Simulasi Tes diharapkan dapat meningkatkan kualitas belajar siswa untuk menghadapi Ujian Sekolah maupun Ujian Nasional.

B. Identifikasi Masalah

Dari latar belakang permasalahan yang muncul dapat diidentifikasi sebagai berikut:

- a) Pengelolaan tes dan hasil tes yang rumit membutuhkan waktu, biaya dan tenaga yang tidak sedikit.
- b) Banyak menggunakan metode pengukuran kemampuan konvensional mengharuskan kehadiran dan tatap muka.
- c) Masih banyak metode pengukuran hasil belajar yang tidak *real-time*.
- d) Penggunaan media evaluasi pembelajaran yang tidak efektif dan responsif.
- e) Belum diketahui kualitas dari Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*, sehingga dibutuhkan pengujian kualitas perangkat lunak yang

meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability* dan *interoperability*.

C. Batasan Masalah

Untuk lebih memfokuskan permasalahan yang akan diteliti, maka perlu adanya batasan masalah sebagai berikut:

1. Pengembangan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* dapat digunakan untuk membantu dalam mengevaluasi pembelajaran siswa yang efektif dan responsif.
2. Pengujian kualitas Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability* dan *interoperability*.

D. Rumusan Masalah

Berdasarkan uraian tentang perlunya pengembangan metode untuk mengukur kemampuan siswa, maka dapat dirumuskan beberapa masalah yaitu:

1. Bagaimana mengembangkan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* untuk *platform Java*?
2. Bagaimana kualitas Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* yang meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability* dan *interoperability*?

E. Tujuan

Tujuan dari penelitian ini adalah menjawab rumusan masalah sebagai berikut:

1. Mengembangkan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* untuk *platform Java*.
2. Mengetahui hasil pengujian kualitas Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* yang meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability*, *interoperability*.

F. Manfaat

Dari penelitian ini penulis mengharapkan dapat memberikan manfaat antara lain:.

1. Manfaat Teoritis

Manfaat teoritis yang yang didapat dalam penelitian ini antara lain:

- a. Sebagai bentuk kontribusi terhadap Universitas baik dalam citra maupun daya tawar terhadap masyarakat luas.
- b. Menambah khasanah keilmuan elektronika khususnya bidang instrumentasi dan telekomunikasi GSM.
- c. Menjadi alternatif media belajar yang melibatkan penggunaan teknologi.
- d. Menambah wawasan tentang pengetahuan elektronika

2. Manfaat Praktis

Manfaat praktis dari pengembangan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* yaitu sebagai berikut:

- a. Menambah khasanah ilmu pengetahuan bidang elektronika dengan menggunakan bahasa pemrograman *Java Micro Edition*.
- b. Memberikan kemudahan dan membantu siswa dalam menghadapi Ujian Nasional.
- c. Siswa dapat belajar Menambah wawasan ilmu dalam hal teknologi informasi dan telekomunikasi bidang aplikasi elektronika.
- d. Guru dapat menjadikan Aplikasi Simulasi Tes sebagai alternatif media belajar siswa sebelum menghadapi Ujian Nasional.

BAB II

KAJIAN PUSTAKA

A. KAJIAN TEORI

1. Definisi Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju (Djati, 2007).

Dalam dunia informatika, aplikasi (*application*) *software* dibentuk untuk menjalankan fungsi tertentu seperti pengolahan kata, angka, presentasi, animasi, multimedia dan sebagainya. Berdasarkan fungsinya aplikasi dapat dikelompokkan ke dalam: aplikasi pengolah kata (*word processing application*) (Djati, 2007).

Dalam taksonomi hasil belajar Benjamin S. Bloom, pendekatan psikologik yakni pada dimensi psikologik aa yang berubah pada peserta didik setelah ia memperoleh pendidikan. Bloom, Englehart, Furst, Hill, & Krathwohl (1956) menjelaskan, pengertian *application* dalam tujuan belajar pada aspek *cognitive domain*, antara lain:

- a. Kemampuan menerapkan materi pelajaran dalam situasi baru.
- b. Kemampuan menetapkan prinsip atau generalisasi pada situasi baru.
- c. Dapat menyusun problema-problema sehingga dapat menetapkan generalisasi.
- d. Dapat mengenali hal-hal yang menyimpang dari prinsip dan generalisasi.
- e. Dapat mengenali fenomena baru dari prinsip dan generalisasi.

- f. Dapat meramalkan sesuatu yang akan terjadi berdasarkan prinsip dan generalisasi.
- g. Dapat menentukan tindakan tertentu berdasarkan prinsip dan generalisasi.
- h. Dapat menjelaskan alasan penggunaan prinsip dan generalisasi.

2. Aplikasi Simulasi

Menerapkan sebuah pembelajaran ke dalam sebuah program diperlukan banyak pemikiran dan pertimbangan yang akurat. Mendefinisikan penerapan simulasi perlu melihat dari suatu objek, benda atau kata yang akan berpengaruh pada persepsi dan paradigma dalam memperlakukan objek.

Menerapkan sebuah simulasi ke dalam perancangan sistem *software* sering diartikan dengan istilah pemodelan. Dengan demikian, menerapkan simulasi adalah gambaran dari realita yang sederhana dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan atau simulasi dapat menggunakan bentuk yang sama dengan realitas. Dalam hal ini penulis merancang sistem ujian/tes secara konvensional kemudian memodelkannya ke dalam aplikasi *mobile*.

Penerapan perancangan sistem ujian/tes tidak terlepas dari rancangan dari praktisi khusus. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas dapat dibagi menjadi 2 (dua) yaitu (Djati, 2007):

3. Metode Pembelajaran Simulasi

Metode pembelajaran simulasi adalah suatu metode pembelajaran yang merupakan perpaduan antara *student centries* dan *teacher centries*. Guru dan siswa secara proporsional bersama-sama mengoptimalkan perannya dalam

proses belajar mengajar. Hal ini sejalan dengan disampaikan oleh Tabrani Tusyan dalam Syaiful Bahri Djamarah (2008: 8) bahwa kegiatan belajar adalah suatu sistem dimana dalam prosesnya kita tidak bisa memisahkan antara peran guru dan peran siswa.

a. Prasyarat Pelaksanaan Metode Pembelajaran Simulasi

Pada prinsipnya, dalam proses belajar mengajar tidak terdapat satu pun metode pembelajaran terbaik. Yang ada adalah metode belajar yang tepat untuk proses belajar tersebut. Artinya, metode pembelajaran sangat dipengaruhi oleh situasi yang kondisi dan saat proses belajar. Dengan demikian, metode simulasi tidak selalu tepat setiap saat untuk digunakan, akan tergantung bagaimana karakteristik dari siswa, guru, materi pembelajaran dan faktor sumber daya yang ada.

Metode pembelajaran simulasi bisa dilaksanakan secara efektif dengan syarat:

- 1) Menurut Depdiknas (2005: 134) dalam buku kumpulan Pembelajaran/Pendampingan, metode simulasi memerlukan ketersediaan bahan dan alat yang memadai untuk melaksanakan simulasi tersebut.
- 2) Kesiapan dari guru untuk mengarahkan siswa dalam melaksanakan simulasi artinya guru memahami betul apa yang harus dilakukan siswa dalam simulasi tersebut. Guru berperan sebagai sutradara yang memberi batasan dan arahan sehingga apa yang disimulasikan tidak keluar dari koridor tujuan pembelajaran (Syaiful Bahri Djamarah, 2006: 92).
- 3) Kesiapan dari siswa untuk melaksanakan simulasi artinya sebelum melaksanakan simulasi siswa sudah memahami apa saja yang harus dilakukan.

4. Simulasi

Simulasi merupakan suatu model pengambilan keputusan dengan mencontoh atau mempergunakan gambar sebenarnya dari suatu sistem kehidupan dunia nyata tanpa harus mengalaminya pada keadaan yang sesungguhnya. Dengan mencontoh atau menduplikasi keadaan yang sebenarnya ini memungkinkan pengambil keputusan untuk melakukan suatu sistem eksperimen terhadap sistem dan prediksi tingkah laku dan hasilnya berdasarkan input berbagai parameter dan aturan. Di samping itu, pengambilan keputusan juga dapat menganalisis berbagai performa sistem dan memilih keputusan yang optimal untuk jangka waktu yang panjang. Simulasi digunakan apabila suatu kasus atau masalah nyata model matematikanya tidak dapat dibentuk atau metode yang tersedia tidak dapat dipakai.

a. Jenis-Jenis Simulasi

Berdasarkan (Hasan M. I., 2002) dalam buku *Pokok-Pokok Materi Pengambilan Keputusan* menjelaskan bahwa simulasi dapat dibedakan menjadi dua, yaitu:

1) Simulasi Analog

Simulasi analog adalah menggantikan lingkungan fisik asli dengan lingkungan fisik tiruan yang lebih mudah untuk dimanipulasi. Simulasi ini mempergunakan representasi fisik untuk menjelaskan karakteristik yang penting dari masalah. Sebagai contoh sistem informasi makro disimulasikan dengan sistem informasi lokal.

2) Simulasi Matematik

Simulasi matematik dapat diartikan meniru sistem dengan model matematik untuk mendapatkan ciri operasi sistem melalui suatu eksperimen. Jika

eksperimen dilakukan secara berulang untuk mempermudah dan mempercepat penyelesaian hitung secara komputasi.

5. Definisi *Software*

Software secara umum dapat diartikan sebagai sekumpulan data-data elektronik yang tersimpan dan diatur oleh komputer yang berupa program atau instruksi untuk menjalankan dan mengeksekusi suatu perintah. *Software* merupakan perangkat yang ada di dalam komputer yang tidak dapat kita *sentuh* dan tidak dapat terlihat bentuk fisiknya. Namun, dapat kita operasikan saat menggunakan komputer. Sedangkan secara khusus, *software* diartikan sebagai sekumpulan perintah/instruksi yang jika dijalankan akan menghasilkan fungsi dan hasil yang diinginkan (Pressman, 2001: 6).

Menurut Agarwal, Tayal dan Gupta (2010: 4), dalam bukunya yang berjudul *Software Engineering and Testing* menyebutkan “*Software is a set of instructions used to acquire inputs and to manipulate them to produce the desired output in terms of functions and performance as determined by the user of the software*”. Kutipan tersebut menyebutkan bahwa *software* adalah sekumpulan instruksi yang digunakan untuk memperoleh masukan (input) dan memanipulasinya. Sehingga menghasilkan keluaran (output) yang diinginkan dalam hal fungsi dan kinerja yang ditentukan oleh pengguna *software*. Beberapa contoh dari *software* adalah Adobe Photoshop, menyediakan fungsi untuk mengolah gambar, Microsoft Word, menyediakan fungsi untuk mengolah dokumen.

Berdasarkan fungsinya, *software* dibedakan menjadi 5, yaitu: *Software* Sistem Operasi (*Operating System*), *Software* Aplikasi (*Application*), *Software*

Bahasa Pemrograman (*Programming Language*), dan *Software* Program bantu (*Utility*). Berikut penjelasan dari masing-masing jenis *software*.

a. *Software* Sistem Operasi (*Operating System*)

Software Sistem Operasi, yaitu program yang berfungsi untuk mengendalikan sistem kerja yang mendasar, sehingga mengatur kerja media input, output, tabel pengodean memori, penjadwalan prosesor, dan lain-lain. Sistem operasi berfungsi sebagai penghubung antara manusia dengan perangkat keras dan perangkat lunak yang digunakan. Adapun fungsi utama sistem operasi adalah:

- 1) Menyimpan program dan aksesnya.
- 2) Membagi tugas di dalam CPU (Computer Processor Unit).
- 3) Mengalokasikan tugas-tugas penting.
- 4) Merekam sumber-sumber data.
- 5) Mengatur memori sistem termasuk penyimpanan, menghapus dan mendapatkan data.
- 6) Memeriksa kesalahan sistem.
- 7) Memelihara keamanan sistem, khusus pada jaringan yang membutuhkan kata sandi (password) dan penggunaan ID.

Contoh *Software* Sistem Operasi, misalnya: Disk Operating System (DOS), *Microsoft* Windows, Linux, dan Unix.

b. *Software* Aplikasi (*Application Software*)

Software Aplikasi adalah perangkat lunak yang dirancang khusus untuk kebutuhan tertentu, misalnya program pengolah kata, mengelola lembar kerja, program presentasi, desain grafis, dan lain-lain.

c. Software Bahasa Pemrograman (*Programming Language Software*)

Software Bahasa Pemrograman yaitu program yang digunakan untuk menerjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman ke bahasa mesin dengan aturan atau prosedur tertentu, agar diterima oleh komputer.

Terdapat 3 level *bahasa* pemrograman, yaitu:

1) Bahasa tingkat rendah (*low level language*)

Bahasa ini disebut juga bahasa mesin (*assembler*), di mana pengodean bahasanya menggunakan kode binary atau angka 0 dan 1.

2) Bahasa tingkat tinggi (*high level language*)

Bahasa ini termasuk dalam bahasa pemrograman yang mudah dipelajari oleh pengguna komputer karena menggunakan bahasa Inggris. Contohnya: *BASIC, COBOL, PASCAL, FORTRAN*.

3) Bahasa generasi keempat (4 GL)

Bahasa pemrograman 4 GL (*Fourth Generation Language*) merupakan bahasa yang berorientasi pada objek yang disebut *Object Oriented Programming* (OOP). Contoh *software* ini adalah: PHP, Java, Visual NET.

6. Tes

Secara harfiah, kata “tes” berasal dari bahasa Perancis Kuno: *testum* dengan arti: “piring untuk menyisihkan logam-logam mulia” (maksudnya dengan menggunakan alat berupa piring itu dapat diperoleh jenis-jenis logam mulia yang nilainya sangat tinggi) dalam bahasa Inggris ditulis dengan *test* yang dalam bahasa Indonesia diterjemahkan dengan “tes”, “ujian” atau “percobaan” dalam bahasa Arab: *imtihan*

Ada beberapa istilah yang memerlukan penjelasan sehubungan dengan uraian di atas. Yaitu, tes adalah alat atau prosedur yang dipergunakan dalam rangka pengukuran dan penilaian. *Testing* berarti, saat dilaksanakannya atau peristiwa berlangsungnya pengukuran dan penilaian. *Tester* artinya orang yang sedang melakukan percobaan (eksperimen). Sedangkan, *testee* (mufrad) dan *testees* (jamak) adalah pihak yang dikenai tes (peserta tes atau peserta ujian), atau pihak yang sedang dikenai pekerjaan (tercoba).

Adapun dari segi istilah, Menurut Bimo Walgito (1987: 87) definisi tes adalah suatu metode atau alat untuk mengadakan penyelidikan yang menggunakan soal-soal, pertanyaan atau tugas-tugas yang lain di mana persoalan-persoalan atau pertanyaan-pertanyaan itu telah dipilih dengan seksama dan telah distandarkan. Menurut Anne Anastasi (1983) dalam jurnal berjudul *Psychological testing: Basic Concept And Common Misconception*, yang dimaksud dengan tes adalah alat pengukur yang mempunyai standar yang obyektif. Sehingga dapat digunakan secara meluas, serta dapat betul-betul digunakan untuk mengukur dan membandingkan keadaan psikis atau tingkah laku individu. Adapun menurut Lee J. Cronbach (1970) dalam bukunya berjudul *Essential of Psychological Testing*, tes merupakan suatu prosedur yang sistematis untuk membandingkan tingkah laku dua orang atau lebih.

Dari definisi-definisi di atas dapat dipahami bahwa dalam dunia evaluasi pendidikan yang dimaksud dengan tes adalah cara (yang dapat dipergunakan) atau prosedur (yang perlu ditempuh) dalam rangka pengukuran dan penilaian di bidang pendidikan yang berbentuk pemberian tugas atau serangkaian tugas, baik berupa pertanyaan-pertanyaan (yang harus dijawab), atau perintah-perintah (yang harus dikerjakan) oleh *testee*. Sehingga atas dasar data yang diperoleh

dari hasil pengukuran tersebut dapat dihasilkan nilai yang melambangkan tingkah laku atau prestasi *testee*.

a. Fungsi Tes

Secara Umum, ada dua macam fungsi tes dalam penerapannya, yaitu:

- 1) Sebagai alat pengukur terhadap peserta didik.
- 2) Sebagai alat pengukur keberhasilan program pengajaran dengan cara melalui tes dapat diketahui seberapa jauh program pengajaran yang telah ditentukan.

b. Penggolongan Tes

Sebagai alat pengukur atau evaluasi, tes dibedakan menjadi beberapa jenis atau golongan dilihat dari segi pandangan dan alasan penggolongan tes dilakukan yaitu:

1) Penggolongan Tes Berdasarkan Fungsinya Sebagai Alat Pengukur Perkembangan/Kemajuan Belajar Peserta Didik.

- a) Tes Seleksi sering dikenal dengan istilah “ujian ringan” atau “ujian masuk”. Tes ini dilaksanakan dalam rangka penerimaan calon siswa baru, di mana hasil tes digunakan untuk memilih calon yang mengikuti tes.
- b) Tes Awal sering dikenal dengan istilah pre-test. Tes jenis ini dilaksanakan dengan tujuan untuk mengetahui sejauh manakah materi atau bahan pelajaran yang akan diajarkan telah dapat dikuasai oleh para peserta didik. Jadi tes awal adalah tes yang dilaksanakan sebelum bahan pelajaran diberikan kepada peserta didik. Karena itu, maka butir-butir soal dibuat dengan kategori mudah.
- c) Tes Akhir sering dikenal dengan post-test. Tes akhir dilaksanakan dengan tujuan untuk mengetahui apakah semua materi pelajaran yang tergolong

penting sudah dapat dikuasai dengan sebaik-baiknya oleh para peserta didik.

- d) Tes Diagnostik adalah tes yang dilaksanakan untuk menentukan secara tepat jenis kesukaran yang dihadapi oleh para peserta didik dalam suatu mata pelajaran tertentu. Dengan diketahuinya jenis-jenis kesukaran yang dihadapi oleh peserta didik itu, maka lebih lanjut akan dapat ditemukan upaya berupa pengobatan yang tepat. Tes Diagnostik bertujuan untuk menemukan jawaban atas pertanyaan “Apakah peserta didik sudah dapat menguasai pengetahuan yang merupakan dasar atau landasan untuk dapat menerima pengetahuan selanjutnya?”.
- e) Tes Formatif adalah tes hasil belajar yang bertujuan untuk mengetahui sejauh manakah peserta didik telah terbentuk sesuai dengan tujuan pengajaran yang telah ditentukan setelah mereka mengikuti proses pembelajaran dalam jangka waktu yang tertentu. Perlu diketahui bahwa istilah “formatif” itu berasal dari kata “form” yang berarti “bentuk”.
- f) Tes Sumatif adalah tes hasil belajar yang dilaksanakan setelah sekumpulan satuan program pengajaran selesai diberikan. Di sekolah, tes ini dikenal dengan istilah Ulangan Umum atau EBTA (Evaluasi Belajar Tahap Akhir) di mana hasilnya digunakan untuk mengisi rapor atau mengisi ijazah (STTB). Tes Sumatif dilaksanakan secara tertulis agar semua siswa memperoleh soal yang sama. Butir-butir soal yang dikemukakan dalam Tes Sumatif ini pada umumnya juga lebih sulit atau lebih berat daripada butir-butir soal Tes Formatif.

2) Penggolongan Tes Berdasarkan Aspek Psikis Yang Ingin Diungkap.

Ditinjau dari segi aspek kejiwaan yang diungkap, tes dapat dibedakan menjadi lima golongan, yaitu:

- a) Tes Intelegensi, yakni tes yang dilaksanakan dengan tujuan untuk mengungkap atau mengetahui tingkat kecerdasan seseorang.
- b) Tes Kemampuan, yaitu tes yang dilaksanakan dengan tujuan untuk mengungkap kemampuan dasar atau bakat khusus yang dimiliki oleh *testee*.
- c) Tes Sikap, yakni salah satu jenis tes yang dipergunakan untuk mengungkap predisposisi atau kecenderungan seseorang untuk melakukan suatu respons tertentu terhadap dunia sekitarnya, baik berupa individu-individu maupun obyek-obyek tertentu.
- d) Tes Kepribadian, yakni tes yang dilaksanakan dengan tujuan mengungkap ciri-ciri khas dari seseorang yang banyak sedikitnya bersifat lahiriah yang sering dikenal dengan istilah pencapaian atau tingkat pencapaian prestasi belajar.
- e) Tes Hasil Belajar yang sering dikenal dengan istilah pencapaian. Yakni, tes yang biasa digunakan untuk mengungkap tingkat pencapaian atau prestasi belajar.

c. Penggolongan Lain-lain

Ditinjau dari segi banyaknya orang yang mengikuti tes, tes dapat dibedakan menjadi dua golongan, yaitu:

- 1) Tes Individual, yakni tes di mana *tester* hanya berhadapan dengan orang *testee* saja.
- 2) Tes Kelompok, yakni tes di mana *tester* berhadapan dengan lebih dari satu orang *testee*.

Ditinjau dari segi waktu yang disediakan bagi *testee* untuk menyelesaikan tes, tes dapat dibedakan menjadi dua golongan, yaitu:

- 1) *Power* tes, yakni tes dimana waktu yang disediakan untuk *testee* menyelesaikan tes tersebut tidak dibatasi.
- 2) *Speed* tes, yakni tes di mana waktu yang disediakan untuk *testee* untuk menyelesaikan tes tersebut dibatasi.

Ditinjau dari segi bentuk responnya, tes dapat dibedakan menjadi dua golongan, yaitu:

- 1) Verbal tes, yakni suatu tes yang menghendaki respons (jawaban) yang tertuang dalam bentuk ungkapan kata-kata atau kalimat baik secara lisan maupun tertulis.
- 2) Non-Verbal tes, yakni tes yang menghendaki respons (jawaban) dari *testee* bukan merupakan ungkapan kata-kata atau kalimat melainkan berupa tindakan atau tingkah laku. Respons yang dikehendaki muncul dari *testee* adalah berupa perbuatan atau gerakan-gerakan tertentu.

Apabila ditinjau dari segi cara mengajukan pertanyaan dan cara memberikan jawabannya, tes dapat dibedakan menjadi dua golongan, yaitu:

- 1) Tes Tertulis, yakni jenis tes di mana *tester* dalam menyajikan butir-butir pertanyaan atau soal dilakukan secara tertulis dan *testee* memberikan jawaban juga secara tertulis.
- 2) Tes Lisan, yakni tes di mana *tester* di dalam mengajukan pertanyaan-pertanyaan atau soalnya dilakukan secara lisan dan *testee* memberikan jawaban secara lisan pula.

7. Tes Pilihan Berganda

Berdasarkan penjelasan Slamet (1988: 59), tes jenis pada pokoknya menghadapkan kepada sejumlah alternatif jawaban, umumnya antara 3 sampai 5 alternatif untuk setiap soal dan tugas siswa adalah memilih salah satu di antara alternatif tersebut berdasarkan sesuatu dasar pertimbangan tertentu (kadang-kadang, sebagai variasi, tidak ditentukan harus memilih satu, tetapi di mana perlu, harus lebih dari yang dipilih).

Stem adalah bagian pokok dari soal yang merumuskan isi soal. *Stem* bisa berbentuk pertanyaan, perintah maupun kalimat tidak sempurna. Alternatif-alternatif jawaban yang menyertainya dinamakan *options* atau diterjemahkan secara langsung, pilihan-pilihan. Alternatif yang benar dinamakan *key* atau kunci, sedangkan alternatif lainnya yang bertujuan mempersulit proses pencapaian jawaban yang benar dinamakan *distractors*, atau secara langsung diterjemahkan, pengganggu-pengganggu/pengecoh.

a. Jenis Tes Pilihan Ganda

Berdasarkan penjelasan Slamet (1988: 60), tes jenis pilihan ganda dibagi menjadi 2, yaitu:

- 1) Dilihat Dari Segi Bentuknya *Stem* Soal-Soal Pilihan ganda, antara lain:
 - a) Pertanyaan, di mana alternatif-alternatif atau optionsnya merupakan kemungkinan-kemungkinan jawabannya.
 - b) Kalimat tidak sempurna di mana options-nya merupakan kemungkinankemungkinan jawabannya.
 - c) Perintah di mana options-nya merupakan kemungkinan-kemungkinan pelaksanaannya, atau

- d) Persoalan yang diungkapkan secara verbal, dengan gambar, grafik, denah dan sebagainya di mana options-nya merupakan pernyataan-pernyataan yang benar atau salah yang berasal dari persoalan gambar, grafik atau denah yang dimaksud.
- 2) Menurut Kemungkinan Cara Menjawabnya, antara lain:
- a) Variasi dengan satu jawaban benar per soal.
 - b) Sejumlah jawaban benar akan tetapi salah satu di antaranya paling benar per soal.
 - c) Satu jawaban yang salah per soal.
 - d) Penjawab menentukan sendiri apakah satu atau lebih options yang benar dan memilih sesuai dengan itu untuk setiap soal.

b. Bentuk Tes Pilihan Ganda

Menurut penjelasan Slamet (1988: 60-62), Menurut penjelasan Slamet (1988: 60-62), tes pilihan ganda dibedakan menjadi lima, yaitu:

- 1) Tes Pilihan Ganda Dengan Menemukan Satu-Satunya jawaban Yang Benar.

Pada jenis ini alternatif kunci memang merupakan satu-satunya jawaban yang benar, baik berdasarkan logika, perhitungan matematika maupun historis atau faktual.

- 2) Tes Pilihan Ganda Dengan Memilih Jawaban yang Terbaik.

Pada jenis ini lebih dari satu *options* mempunyai kebenaran yang bertingkat-tingkat dan *testee* diharapkan menemukan *options* yang paling benar, paling lengkap atau paling kena untuk menjawab persoalan yang dikemukakan.

3) Tes Pilihan Ganda Dengan Memilih Lebih Dari Satu Jawaban Benar.

Pada jenis ini setiap soal mempunyai satu atau lebih jawaban benar dan terserah kepada *testee* untuk menentukan beberapa dan yang mana *options* yang merupakan *key* untuk setiap soal.

Mudah dibayangkan bahwa variasi ini adalah salah satu variasi yang tersulit sehingga kemungkinan kesalahan jauh lebih besar, jika bahan soal tidak dikuasai, setiap alternatif harus benar-benar diketahui atau dikuasai siswa sebelum ia bisa memutuskan apakah sesuatu *options* termasuk kunci atau tidak; menjawab dengan menerka saja akan membuat keadaan lebih parah sebab ia akan dianggap membuat kesalahan setia ia tidak memilih alternatif yang merupakan kunci.

4) Tes Pilihan Ganda Dengan Soal Kalimat Negatif

Pada jenis ini *stem* merupakan kalimat negatif dan siswa diharapkan menemukan *options* yang tidak termasuk dalam rumusan berdasarkan sesuatu dasar fikiran.

5) Tes Pilihan Ganda Yang Menggunakan Gambar, Bagan, Peta, Dan Lain-Lain.

Pada jenis ini persoalan tidak dirumuskan dalam bentuk suatu kalimat yang merupakan *stem*, akan tetapi berupa gambar, bagan, peta, dan lain-lain. Siswa pertama-tama harus mengerti persoalan dari dalam gambar, bagan, peta, grafik dan lainnya itu sebelum bisa menjawab pertanyaan-pertanyaan.

c. Kelebihan Dan Kekurangan Tes Soal Pilihan Ganda

Dalam penyajian soal dalam bentuk pilihan ganda untuk evaluasi pembelajaran, terdapat beberapa kelebihan dan kelemahan dalam implementasi ke dalam media pembelajaran (Slamet, 1988: 63-64; Marsh, 2005: 1155), yaitu:

1) Kelebihan Tes Pilihan Ganda

- a) Lebih fleksibel dan efektif.
- b) Mencakup hampir seluruh bahan pelajaran.
- c) Tepat untuk mengukur penguraian informasi, perbendaharaan kata-kata, pengertian-pengertian, aplikasi prinsip, rumus, serta kemampuan untuk menginterpretasikan data.
- d) Dapat juga untuk mengukur kemampuan siswa dalam hal membuat tafsiran, melakukan pemilihan, mendiskriminasikan, menentukan pendapat atas dasar alasan tertentu, dan menarik kesimpulan.
- e) Koreksi dan penilaiannya mudah.
- f) Obyektif.
- g) Dapat dipakai berulang-ulang.
- h) Meningkatkan kinerja siswa dalam ujian.

2) Kelemahan Tes Pilihan Ganda

- a) Sulit serta membutuhkan waktu yang dalam dalam menyusun soal soalnya.
- b) Tidak dapat dipakai untuk mengukur kecakapan siswa dalam mengorganisasikan bahan.
- c) Menghabiskan waktu belajar untuk tes.
- d) Mengambil catatan atau mencari materi yang berbeda daripada mengerjakan soal essay.
- e) Dapat mengakibatkan misinformasi (missed information).

8. Aplikasi Berbasis *Handphone* (*Mobile Application*)

Teknologi *smartphone* OS (*Operating System*) *Java* saat ini banyak dikembangkan oleh perusahaan-perusahaan besar. Yaitu, Sony Ericsson, Nokia, Motorola, LG, Siemens. *Smartphone* OS *Java* telah didesain bagi alat komunikasi personal yang mendukung komunikasi tanpa kabel (*wireless*) seperti pada *mobile phone* maupun PDA (*Personal Digital Assistance*). Teknologi *wireless* yang terpasang pada *handphone* memungkinkan terhubung komunikasi secara *online* dengan kecepatan transfer 54 MBps. Maka, kemampuan *device* tersebut dapat mendukung bagi pengembang aplikasi *online* dengan teknologi *mobile application*.

Mobile application adalah proses pengembangan aplikasi untuk perangkat genggam seperti PDA, asisten digital perusahaan atau telepon genggam. Aplikasi ini sudah ada pada telepon selama manufaktur, atau diunduh oleh pelanggan dari toko aplikasi dan dari distribusi perangkat lunak *mobile platform* yang lain.

Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, desain dan *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop (JENI, 2012: 1)

Adapun beberapa karakter perangkat *mobile* menurut JENI (2012: 1-2; Baskara, 2012), antara lain:

a. Ukuran Kecil

Perangkat *mobile* memiliki ukuran yang kecil. Pengguna menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas.

b. Memory yang terbatas

Perangkat *mobile* juga memiliki memori yang kecil. Yaitu, *primary* (RAM) dan *secondary* (*disk*). Pembatasan ini adalah salah satu faktor yang memengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah memori, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal.

c. Daya proses yang terbatas

Sistem *mobile* tidaklah setangguh aplikasi *desktop*. Ukuran, teknologi dan biaya adalah beberapa faktor yang memengaruhi status dari sumber daya ini. Seperti *harddisk* dan RAM (*Random Access Memory*) dapat menemukan mereka dalam ukuran yang tepat dengan sebuah kemasan kecil.

d. Mengonsumsi daya yang rendah

Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan oleh baterai-baterai.

e. Kuat dan dapat diandalkan

Karena perangkat *mobile* selalu dibawa ke mana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

f. Konektivitas yang terbatas

Perangkat *mobile* memiliki *bandwidth* rendah dan beberapa hari bahkan tidak tersambung. Kebanyakan dari mereka menggunakan dan memanfaatkan koneksi *wireless*.

9. Pengenalan Java

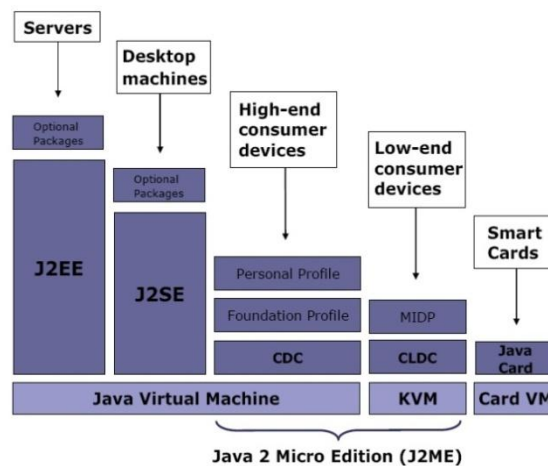
Java adalah nama sebuah bahasa pemrograman yang diciptakan oleh *Sun Microsystems*, sebuah perusahaan besar di Amerika Serikat. *Java* dibuat pada tahun 1991 oleh James Gosling yang awalnya *compiler* tersebut diberi nama *Oak*, di mana untuk menghormati pohon yang ada di luar jendela *Gosling*. Kemudian namanya diubah ke *Java* karena telah ada sebuah bahasa yang diberi nama *Oak*. Motivasi sesungguhnya dari *Java* adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai *platform* yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan *Java* adalah sebuah *remote control* yang diberi nama *Star 7* (Soesilo, Hartati, & Suharto, 2005: 1-3).

Pada saat yang sama *World Wide Web* dan *Internet* berkembang sangat cepat. *Gosling* menyadari bahwa *Java* dapat digunakan untuk pemrograman *Internet*. Dengan keluarnya versi 1.2, *platform Java* akhirnya diklasifikasi menjadi beberapa edisi. Yaitu, *The Standard Edition (Java SE)*, *Enterprise Edition (Java EE)*, *Micro Edition (Java ME)* dan *JavaCard API*.

Java menurut definisi dari *Sun* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan (Shalahuddin, 2011: 1).

Tabel 1. *Platform Java* (Taptozani, 2009: 10)

<i>Java SE – Java Standard Edition</i>	Aplikasi Desktop
<i>Java EE – Java Enterprise Edition</i>	Aplikasi enterprise dengan focus pada pengembangan sisi web server, termasuk server, <i>JSP</i> , <i>EJB</i> dan <i>XML</i>
<i>Java ME – Java Micro Edition</i>	Perangkat Mobile
<i>JavaCard</i>	<i>Smart Cards</i>



Gambar 2. Platform Java (Taptozani, 2009: 10)

a. J2SE (Java 2 Second Edition)

Java 2 Second Edition atau yang dikenal dengan J2SE adalah salah satu platform dari bahasa pemrograman java yang diimplementasikan untuk aplikasi desktop. J2SE terdiri dari *Java Runtime Environment* (JRE) yang terdiri dari banyak *library* standar. Sedangkan, untuk *Java Development Kit* (JDK) adalah JRE yang ditambahkan development *Tools & API*. Untuk membuat aplikasi berbasis *user interface*, telah terdapat paket *Abstract Window Toolkit* (AWT) yang bergantung dari sistem operasi yang digunakan. *User interface* pada linux akan berbeda dengan *user interface* pada *open solaris*. Untuk membangun aplikasi *user interface* yang tidak bergantung pada sistem operasi yang digunakan dapat menggunakan paket *Swing*. Tampilan *user interface* akan selalu sama walaupun berpindah-pindah sistem operasi.

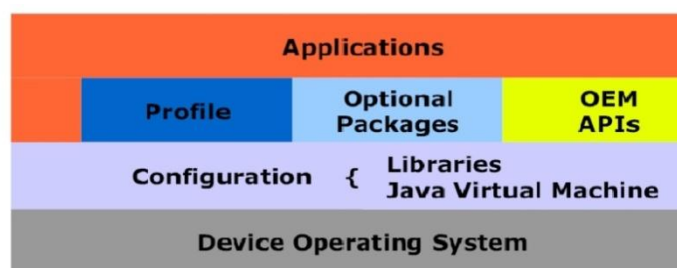
b. JME (Java Micro Edition)

Java ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat *mobile*. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan *bandwidth* jaringan yang rendah. Dengan perkembangan perangkat mobile dari telepon, PDA, kotak

permainan hingga peralatan rumah. Java menyediakan suatu lingkungan yang portabel untuk mengembangkan dan menjalankan aplikasi pada perangkat ini (Taptozani, 2009: 26-27).

Program *Java ME*, seperti semua program Java adalah diterjemahkan oleh *Virtual Machine (VM)*. Program-program tersebut dikompile ke dalam *bytecode* dan diterjemahkan oleh *Java Virtual Machine (JVM)*. Ini berarti, bahwa program-program tersebut tidak berhubungan langsung dengan perangkat keras. *Java ME* menyediakan suatu *interface* yang sesuai dengan perangkat keras. Aplikasi-aplikasi tersebut cukup dikompile sekali dan mampu menjalankan pada mesin yang berbeda.

Inti dari *Java ME* terletak pada *configuration* dan *profile-profile*. Suatu *configuration* mengembangkan lingkungan *runtime* dasar dari suatu sistem *Java ME*. Ia mengembangkan *core library*, *virtual machine*, fitur keamanan dan jaringan.



Gambar 3. Arsitektur Java ME (Taptozani, 2009:12)

Sebuah *profile* memberikan *library* tambahan untuk suatu kelas tertentu pada sebuah perangkat. *Profile-profile* menyediakan *user interface (UI) API*, *persistence*, *messaging library* dan sebagainya. Satu set *library* tambahan atau *package* tambahan menyediakan kemampuan program tambahan. Pemasukan *package* tergantung pada kemampuan sebuah perangkat. Sebagai contoh,

beberapa perangkat MIDP tidak memiliki *Bluetooth built-in*, sehingga *Bluetooth API* tidak disediakan dalam perangkat ini.

1) **Configuration**

Suatu *configuration* menggambarkan fitur minimal dari lingkungan lengkap *Java runtime*. Untuk menjamin kemampuan portabilitas dan interoperabilitas yang optimal di antara berbagai macam perangkat yang dibatasi sumber daya (*memory, processor, koneksi yang dibatasi*), *configuration* tidak menggambarkan fitur tambahan. Suatu *configuration Java ME* menggambarkan suatu komplemen yang minimum dari teknologi *Java*. Merupakan tugas *profile-profile* untuk menggambarkan tambahan *library* untuk suatu kategori perangkat tertentu. *Configuration* menggambarkan (Taptozani, 2009: 12):

- a) Subset bahasa pemrograman *Java*.
- b) Kemampuan Java Virtual Machine (JVM)
- c) Core platform libraries
- d) Fitur sekuriti dan jaringan.

JME terdiri dari dua *configuration* yaitu:

- a) *Connected Limited Device Configuration (CLDC)*

Digunakan pada *device* yang terbatas dalam hal memori yang kecil dan prosesor yang tidak begitu cepat. Hanya terdiri dari sebagian kecil *class library* J2SE seperti *java.lang* dan *java.util*. Kemudian ditambahkan paket untuk aplikasi *mobile* *javax.microedition.io*. CLDC banyak digunakan pada handphone dan PDA (*Personal Digital Asistants*).

- b) *Connected Device Configuration (CDC)*

Berisikan semua dari *Java Virtual Machine (JVM)*, dan ditambahkan paket untuk aplikasi *mobile*. Sehingga membutuhkan memori yang lebih besar dan

prosesor yang lebih cepat dari CLDC. CDC banyak digunakan pada internet tv, car tv dan nokia *communicator*.

2) **Profile**

Suatu *profile* menggambarkan set-set tambahan dari *API* dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara, *configuration* menggambarkan *library* yang penting untuk membuat aplikasi-aplikasi efektif. *Library* ini memasukan *user interface*, jaringan dan penyimpanan *API*.

Profile merupakan *class library* tambahan pada *configuration* yang digunakan untuk pengelompokan spesifikasi tertentu (*domain-specific*), dan digunakan untuk mendukung paket-paket opsional agar dapat mengakses *virtual machine*. *Profile* merupakan kebalikan dari *configuration*, yaitu mengatur hal-hal yang spesifik untuk sebuah *device*. Terdapat dua buah profil dalam JME yaitu: MIDP (*Mobile Device Information Profile*) dan *Foundation Profile* (FP) (Taptozani, 2009: 13-15).

a) MIDP (*Mobile Device Information Profile*)

MIDP atau *Mobile Information Device Profile*. Yaitu *profile* yang menyediakan *library* Java untuk implementasi dasar antarmuka (GUI), jaringan (*networking*), basis data (*database*), penyimpanan tetap (*persistant storage*), daur hidup aplikasi (*Application life cycle*), dan *timer* (*Personal Information Management*). MIDP dirancang khusus untuk *wireless phone* dan *pager* serta dibangun berdasarkan CLDC. Saat ini terdapat MIDP 1.0 dan MIDP 2.0 . Fitur tambahan MIDP 2.0 dibanding dengan MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan *tone*, *tone sequence*, dan *file* WAV walaupun tanpa adanya *Mobile Media API* (MMAPI).

Spesifikasi *MIDP* menggambarkan suatu perangkat *MIDP* yang memiliki karakteristik-karakteristik ini sebagai berikut (Taptozani, 2009: 14; Baskara, 2012):

(1) Tampilan:

- Ukuran Layar: 96x54-400x240 (3")
- Kedalaman tampilan: 1-bit– 16-bit (155 ppi)
- Ketajaman pixel: sekitar 1:1– 1:16

(2) Masukan:

Satu atau lebih mekanisme *user-input*: satu *keyboard*, dua *keyboard*, atau *touch screen*.

(3) Memori:

- 25 kilobytes of non-volatile memory untuk implementasi MIDP – 32 Megabytes of non-volatile memory untuk implementasi MIDP .
- 8 kilobytes of non-volatile memory for application-created persistent data – 10 Megabytes of non-volatile memory for application-created persistent data.
- 128 kilobytes of volatile memory for the Java runtime (e.g., the Java heap) – 2 Megabytes of volatile memory for the Java runtime.

(4) Jaringan:

Dua jalur, *wireless*, *bandwidth* terbatas – *bluetooth* (*Book Access Profile* (PBAP) 1.1), *Dual SIM*, GSM: 900 MHz-1800 MHz.

(5) Sound:

Kemampuan untuk memainkan nada - Kamera 2 MP, resolusi video kamera utama: QCIF (176 x 144), laju bingkai video kamera: 10 fps,

perbesaran video kamera: 4 x, fitur perekaman video: Perbesaran video, format perekaman video: 3GP/H.263.

10. Bahasa Pemrograman PHP (*Hypertext Preprocessor*)

PHP merupakan singkatan dari *HypertextPreprocessor* yaitu bahasa pemrograman *webserver-side* yang bersifat *opensource*. *PHP* merupakan *script* yang berintegrasi dengan *HTML* dan berada pada *server* (*serversideHTML embedded scripting*). *PHP* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *scriptPHP* dieksekusi oleh *server* di mana *script* tersebut diletakkan (Anhar, 2010: 3). Kode *PHP* juga dapat berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan (Kadir, 2008: 2).

Menurut Valade (2004: 16), *PHP* dapat berkembang sangat cepat karena memiliki banyak kelebihan, antara lain sebagai berikut:

- a. Cepat, karena kode *PHP* tertanam dalam *HTML*, sehingga memiliki respons yang lebih cepat;
- b. Tidak mahal, karena *PHP* tersedia secara gratis dan bebas digunakan oleh siapa saja;
- c. Mudah digunakan, *PHP* berisi banyak fitur-fitur khusus dan fungsi yang dibutuhkan untuk membuat halaman *web* dinamis;
- d. Dapat berjalan di berbagai sistem operasi, *Windows*, *Linux*, *Mac OS*, dan turunan dari *Unix*;

- e. Dukungan tersedia secara luas, banyak pengguna yang memberikan dukungan secara gratis melalui email dan forum;
- f. Aman, karena kode *PHP* tidak terlihat saat dieksekusi dan hanya menghasilkan *HTML*;
- g. Dirancang untuk mendukung *database*, *PHP* memang dirancang untuk berinteraksi dengan *database*;
- h. *Open source*, lisensi *open source* memungkinkan *programmer* untuk memodifikasi *software PHP* agar sesuai dengan kebutuhan.

Dari seluruh penjelasan di atas maka dapat diambil kesimpulan bahwa *PHP* adalah bahasa pemrograman *script* yang berjalan di *server* dan berfungsi untuk menghasilkan sebuah halaman *web* dinamis. *PHP* dirancang untuk terhubung dengan *database*, salah satunya *database* yang kompatibel dengan *PHP* adalah *MySQL*. Banyak kelebihan yang dimiliki *PHP*, salah satunya adalah cepat, aman dan murah. *PHP* juga bersifat *open source* sehingga pengguna dapat memodifikasi *PHP* agar sesuai kebutuhan.

11. *MySQL*

Salah satu kebutuhan dari *software* yang akan dibuat adalah menyimpan data, dimana semua data dari jadwal yang disusun akan disimpan pada *database*, baik sebelum jadi maupun setelah jadi. *Database* merupakan tempat penyimpanan data (angka dan huruf) dari sebuah sistem. Sedangkan penulis menggunakan salah satu system *database* yang ada yaitu *MySQL*. Seperti yang dikutip dari buku *Beginning MySQL*, "*MySQL is a relational database management system (RDBMS), allows you to manage data and support applications that rely on access to MySQL data*" (Sheldon & Moes, 2005: 1).

Dijelaskan bahwa *MySQL* merupakan sebuah sistem manajemen *database* yang ter-relasi yang dapat digunakan untuk mendukung aplikasi yang membutuhkan *database*.

Penulis memilih *MySQL* karena pada software yang akan dibuat melibatkan struktur hirarki antara satu variabel dengan variabel lainnya. Seperti hubungan antara guru dengan pelajaran, hari dengan jam pelajaran, serta variabel lain yang harus dipenuhi guna mendukung pembuatan software ini. Berikut ini adalah kelebihan dari *MySQL* menurut (Sheldon & Moes, 2005: 8) dalam bukunya yang berjudul *Beginning MySQL*, yaitu:

1. *MySQL* adalah *database* yang memiliki kecepatan yang tinggi dalam melakukan pemrosesan data, dapat diandalkan, dan mudah digunakan serta mudah dipelajari.
2. *Fully Multi Threaded* dengan *kernel thread* artinya adalah bisa dengan mudah mempergunakan *multiple* CPU bila ada.
3. *API (Application Programming Interface)* dengan C, C++ , Eiffel, Java, Perl, PHP, Python dan Tel
4. *MySQL* mendukung banyak bahasa pemrograman seperti C, C++, Perl, Python, Java, dan PHP. Selain itu, dengan bantuan ODBC, *MySQL* juga mampu berinteraksi dengan berbagai pemrograman visual seperti Delphi, Visual Basic, Java, dan sebagainya.
5. *MySQL* dapat melakukan koneksi dengan client menggunakan *protocol* TCP/IP, Unix socket (Unix), atau Named Pipes (NT).
6. *MySQL* dapat menangani data dengan skala yang sangat besar dengan jumlah *record* mencapai lebih dari 50 juta, menampung 60 ribu tabel, dan juga bisa menampung 5 milyar baris data.

7. Dalam hal relasi antartabel *MySQL* menerapkan metode *onesweep multijoin*, sehingga sangat efisien dalam mengelola informasi yang kita minta dari beberapa tabel sekaligus
8. *Multiusers*, yaitu dalam satu database server pada *MySQL* dapat diakses oleh beberapa user dalam waktu yang sama tanpa mengalami konflik atau kemacetan sistem.
9. *Security* yang dimiliki database *MySQL* dikenal baik, karena memiliki lapisan sekuritas seperti level *subnetmask*, nama *host* dan izin akses *user* dengan sistem perizinan yang khusus serta *password* yang dimiliki setiap *user* dalam bentuk data terenkripsi

12. Software Quality

Agarwal, Tayal dan Gupta (2010: 89), menjelaskan *software quality* sebagai kesesuaian pada semua perangkat lunak yang dibangun berkaitan dengan fungsi perangkat lunak, standar pembangunan perangkat yang terdokumentasi, dan karakteristik yang ditujukan oleh perangkat lunak. Definisi ini menekankan pada 3 hal, yaitu:

- a. Kebutuhan perangkat lunak adalah dasar ukuran kualitas perangkat lunak, jika perangkat lunak tidak sesuai dengan kebutuhan yang ditentukan, maka kualitasnya pun berkurang.
- b. Jika menggunakan sebuah standar untuk pembangunan perangkat lunak, maka perangkat lunak dianggap kurang berkualitas jika tidak memenuhi standar tersebut.

- c. Sering kali ada kualitas yang secara langsung tertulis, seperti kemudahan penggunaan dan pemeliharaan yang baik. Sehingga kualitas perangkat lunak dipertanyakan jika tidak memenuhi kebutuhan ini.

Kualitas perangkat lunak atau *software quality* didefinisikan sebagai kesesuaian terhadap kebutuhan performa dan fungsionalitas, standar pengembangan yang terdokumentasi, serta karakter implisit dari sebuah perangkat lunak yang dikembangkan secara profesional (Agarwal, Tayal dan Gupta, 2010: 89). Sebuah perangkat lunak perlu dijaga untuk keperluan sebagai berikut (Shalahuddin & Rosa, 2011: 209):

- a. Agar dapat “*survive*” bertahan hidup di dunia bisnis perangkat lunak.
- b. Dapat bersaing dengan perangkat lunak yang lain.
- c. Penting untuk pemasaran global (global marketing).
- d. Mengefektifkan biaya agar tidak banyak membuang perangkat lunak karena kegagalan pemasaran atau kegagalan produksi.
- e. Mempertahankan pelanggan (customer) dan meningkatkan keuntungan.

13. UML (*Unified Modeling Language*)

“The Unified Modeling Language (UML) is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system.” Dijelaskan oleh James Rumbaugh, Ivar Jacobson, dan Grady Booch (1999: 3) bahwa UML adalah metode permodelan visual yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak.

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana merancang dan membuat *software* berorientasi obyek. Karena

UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi obyek, maka semua elemen dan diagram berbasis pada paradigma *object oriented*. Pengertian lain menjelaskan bahwa UML (*Unified Modeling Language*) adalah sebuah bahasa standar untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model. Tetapi, tidak menyampaikan sesuatu dan kapan model yang harus dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *objek-oriented database*.

a. Use Case Diagram

“The purpose of the use case view is to list the actors and use cases and show which actors participate in each use case” James Rumbaugh, Ivar Jacobson, dan Grady Booch (1999: 26). Tujuan dari gambaran *use case diagram* adalah untuk menjelaskan daftar aktor dan aktivitas penggunaan serta aktor berpartisipasi dalam aktivitas setiap *use case*.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dengan sistem.

Konsep dasar pemodelan *use case* meliputi: *use case*, *actor*, relasi, diagram aktivitas dan diagram *use case*. Aktor merupakan bagian dari *use case* yang bertindak sebagai subyek (pelaku) dalam suatu proses. *Use case* adalah proses-proses yang terjadi dalam suatu *software*. *Use case* juga menggambarkan apa

yang sedang dilakukan oleh seorang *actor*. Relasi menggambarkan hubungan antara *actor* dan *use case*.

b. Activity Diagram

“An activity diagram shows the procedural steps involved in performing a highlevel operation”. Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis diagram aktivitas dapat digunakan untuk menunjukan aliran kerja bisnis. Diagram ini dapat digunakan untuk menggambarkan aliran kerja (*flow of events*) dalam *use case* James Rumbaugh, Ivar Jacobson, dan Grady Booch (1999: 310).

c. Sequence Diagram

“Sequence diagrams show the behavioral view of collaborations explicitly, including the time sequencing of messages and an explicit representation of method activations” James Rumbaugh, Ivar Jacobson, dan Grady Booch (1999: 309). Dijelaskan bahwa *sequence diagram* digunakan menunjukkan pandangan perilaku kolaborasi secara eksplisit, termasuk waktu urutan pesan dan representasi eksplisit dari metode aktivasi.

d. Class Diagram

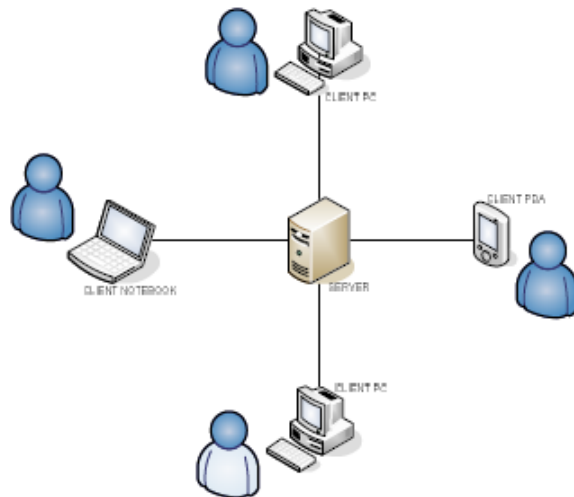
“A class diagram is a graphic presentation of the static view that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships” James Rumbaugh, Ivar Jacobson, dan Grady Booch (1999: 190). Dijelaskan bahwa *class diagram* digunakan untuk menjelaskan presentasi grafis dari pandangan statis yang menunjukkan koleksi dari deklaratif (statis) elemen model, seperti kelas, jenis, isi serta hubungan.

14. Aplikasi *Client-Server*

“This model is based on the distribution of functions between two types of independent and autonomous entities: Server and Client. A Client is any process that request specific services from server processes. A Server is process that provides requested services for Clients. Or in other words, we can say “A client is defined as a requester of services and a server is defined as the provider of services.” A single machine can be both a client and a server depending on the software configuration. Client and Server processes can reside in same computer or in different computers linked by a network.” (Yadav & Singh, 2009: 4)

Menurut Subhash Chandra dan Sanjay Kumar, model *Client-Server* didasarkan pada distribusi fungsi antara dua jenis entitas independen dan otonom. Yaitu, *server* dan *client*. *Client* adalah proses apa pun yang berupa permintaan layanan dari proses *server*. *Server* adalah proses yang menyediakan layanan yang diminta untuk *client*. Atau dengan kata lain, kita dapat mengatakan, "Seorang klien didefinisikan sebagai pemohon layanan dan *server* didefinisikan sebagai penyedia jasa."

Client-server merupakan arsitektur komputasi yang memisahkan antara *client* dengan *server*, dan kebanyakan arsitektur ini diimplementasikan pada jaringan komputer. Namun, bisa juga diimplementasikan pada perangkat lain seperti juga *handphone* dengan *handphone*, ataupun *handphone* dengan komputer. Setiap klien atau *server* yang terhubung ke jaringan disebut sebagai *node*. Umumnya, arsitektur ini memiliki 2 tipe *node*. Yaitu, *client* dan *server*. Tipe ini biasanya disebut pula *two-tier*. Dengan arsitektur ini, memungkinkan *device* untuk melakukan *sharing files* dan *resources* (Taptozani, 2009: 33). Berikut adalah gambar arsitektur umum *client-server*:



Gambar 4. Arsitektur Umum *Client Server* (Taptozani, 2009: 33)

Setiap *instance* dari *software* pada klien dapat mengirimkan *request* kepada satu atau beberapa *server* yang terhubung kepadanya. *Server* kemudian menerima *request* tersebut, memrosesnya, setelah itu mengirimkan informasi yang dihasilkan kepada klien. Meskipun garis besar pola *request* dan *response*-nya tetap sama. Beberapa penerapan arsitektur *client server*, yaitu pada *web browser* sebagai klien, serta *web server*, *database server*, dan *mail server* sebagai *server*.

a. Karakteristik *Client-Server*

Secara umum karakteristik dari *client* adalah sebagai berikut (Taptozani, 2009: 34):

- 1) Aktif.
- 2) Menginisialisasi *request*.
- 3) Menunggu balasan *request*.
- 4) Umumnya terhubung dengan sedikit *server* dalam satu waktu.
- 5) Umumnya berinteraksi secara langsung dengan *end-user* menggunakan *GUI* (*Graphical User Interface*).

Sedangkan karakteristik *server* secara umum yaitu (Taptozani, 2009: 34):

- 1) Pasif.
- 2) Menunggu *request* dari *client/user*.
- 3) Seketika terdapat *request* yang masuk kepadanya, *server* akan langsung memproses dan mengirimkan hasil informasinya kepada klien yang memintanya.
- 4) Umumnya, menerima koneksi dari klien dalam jumlah yang besar
- 5) Umumnya, tidak berinteraksi secara langsung dengan *end-user*.
- 6) Dapat bersifat *stateless* ataupun *stateful*.

b. Keuntungan dan Kerugian Arsitektur *Client Server*

Keuntungan dari penerapan arsitektur *client-server* antara lain (Taptozani, 2009: 34):

- 1) Semua data disimpan pada *server*, sehingga manajemen sekuritas penuh berada pada *server*. Hal ini, menjadi baik dibandingkan dengan manajemen sekuritas data yang tersebar, *server* menjamin siapa-siapa saja yang berhak mengakses data.
- 2) Dikarenakan memiliki *storage* terpusat, proses *update* data menjadi mudah untuk dilakukan daripada melakukan *update* data yang tersebar. Pada kasus *peer to peer*, update data dilakukan di setiap *peer*. Hal ini, tentunya sangat memakan waktu dan rentan terjadinya inkonsistensi data.

Sedangkan kerugian dari penerapan arsitektur *client-server* antara lain (Taptozani, 2009: 35):

- 1) Kongesti trafik menjadi masalah utama arsitektur *client-server*. Semakin banyak jumlah *request client* secara simultan, dapat mengakibatkan *server* dan jaringan kelebihan beban.

- 2) Jika *server* mengalami kegagalan, maka *request* dari klien tidak dapat dipenuhi. Hal ini, berlaku untuk semua klien yang terkoneksi dengan *server*.

15. Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) atau yang sering disebut dengan model proses pengembangan perangkat lunak merupakan suatu proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak (Shalahuddin & Rosa, 2011: 28).

Menurut penjelasan Rosa dan Shalahuddin, SDLC merupakan suatu tahapan proses pengembangan perangkat lunak. Pengembangan perangkat lunak diharuskan melalui semua tahapan yang ada pada SDLC agar hasil dari proses pengembangan tersebut berkualitas. Setiap model SDLC memiliki tahapan-tahapan tersendiri. Setiap model SDLC memiliki kelebihan dan kekurangan tersendiri. Berikut beberapa model SDLC yang banyak digunakan oleh para pengembang:

a. Model *Waterfall*

Model *waterfall* sering disebut juga model sekuensial linier dimana setiap tahapan yang ada dilakukan secara bertahap. Tahapan model *waterfall* meliputi analisis kebutuhan, desain, pengkodean, pengujian, dan pemeliharaan. Tahapan-tahapan dari model *waterfall* dikerjakan secara berurutan.

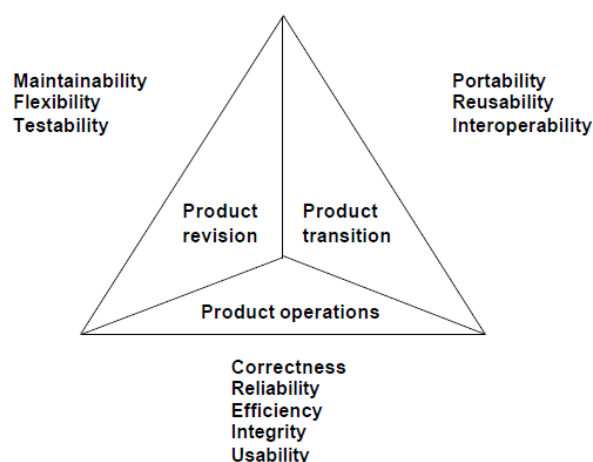
Model *waterfall* memiliki kelebihan dan kekurangan. Dengan segala kelebihan dan kekurangannya model ini merupakan model SDLC yang paling sederhana dan sangat cocok bagi pengembangan perangkat lunak dengan

spesifikasi yang tidak berubah ubah. Berdasarkan pendapat tersebut dapat dikatakan bahwa dalam pengembangan menggunakan model *waterfall* tahap analisis kebutuhan menjadi kunci proses pengembangan. Analisis kebutuhan harus matang dan mencakup seluruh elemen dari komponen perangkat lunak (Shalahuddin & Rosa, 2011: 26-27).

Model SDLC yang cocok digunakan dalam penelitian ini adalah model *waterfall*. Tahapan model *waterfall* meliputi analisis kebutuhan, desain, pengkodean, pengujian, dan pemeliharaan. *Waterfall* merupakan model SDLC yang paling sederhana dan sangat cocok dalam pengembangan perangkat lunak dengan spesifikasi yang tidak berubah.

16. *McCall's Quality Factors*

McCall, Richard, dan Walter dalam Pressman (2000) merumuskan sebelas faktor yang ada dalam *software quality* dikategorikan ke dalam tiga aspek penting dari produk *software*. Yaitu, karakteristik operasional (*product operation*), kemampuannya untuk mengalami perubahan (*product revision*) dan adaptasi dengan lingkungan baru (*product transition*).



Gambar 5. *McCall's Software Quality Factors* (Pressman R. S., 2000)

Berdasarkan gambar di atas, McCall menghubungkan *quality factor* dengan *metric*. Berikut adalah gambar hubungan antara *quality factor* dengan *metric*:

Quality factor	Software quality metric	Correctness	Reliability	Efficiency	Integrity	Maintainability	Flexibility	Testability	Portability	Reusability	Interoperability	Usability
Auditability					X			X				
Accuracy			X									
Communication commonality												X
Completeness		X										
Conciseness				X		X	X					
Consistency		X	X			X	X					
Data commonality												X
Error tolerance			X									
Execution efficiency				X								
Expandability							X					
Generality							X			X	X	X
Hardware independence									X	X		
Instrumentation					X	X		X				
Modularity			X			X	X	X	X	X	X	
Operability				X								X
Security					X							
Self-documentation						X	X	X	X	X		
Simplicity			X			X	X	X				
Software system independence									X	X		
Traceability		X										
Training												X

Gambar 6. Hubungan McCall's *Quality Factor* dengan *Metric* Pengujian (Pressman R. S. , 2000)

Berikut ini adalah kategori produk menurut McCall:

a. *Product Operation*

- 1) *Correctness* merupakan karakteristik yang berakaitan dengan bagaimana program mampu memenuhi spesifikasi dan tujuan yang ingin dicapai oleh pengguna (Pressman, 2001: 509). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Completeness*

Karakteristik yang menunjukan tingkat di mana implementasi penuh dari fungsi yang dibutuhkan telah dicapai.

b) *Consistency*

Karakteristik yang menunjukkan penggunaan satu bentuk yang seragam pada rancangan seluruh sistem.

c) *Traceability*

Karakteristik program yang menunjukkan kemampuan untuk melacak representasi desain atau program dapat kembali ke tampilan awal.

- 2) *Reliability* merupakan karakteristik yang berkaitan dengan bagaimana sebuah program mampu beroperasi dalam sebuah kondisi yang menuntut presisi tertentu (Pressman, 2001: 509). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Accuracy*

Karakteristik yang menunjukkan kemampuan *software* untuk memberikan hak kontrol atau hasil dan efek (perhitungan) dengan tingkat presisi yang standar atau yang telah ditetapkan.

b) *Consistency*

Karakteristik yang menunjukkan penggunaan satu bentuk yang seragam pada seluruh rancangan sistem.

c) *Error Tolerance*

Karakteristik yang menunjukkan indikasi kesalahan proses ketika program sedang berjalan.

d) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

e) *Simplicity*

Karakteristik yang menunjukkan tingkat sejauh mana *user* dapat memahami program tanpa mengalami kesulitan.

- 3) *Efficiency* merupakan karakteristik yang berkaitan dengan jumlah sumber daya komputer yang digunakan serta kode yang diperlukan dalam program untuk menjalankan setiap fungsinya (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Conciseness*

Karakteristik yang menunjukkan bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang relatif ringkas.

b) *Execution Efficiency*

Karakteristik yang menunjukkan bahwa eksekusi program berjalan secara efektif.

c) *Operability*

Karakteristik yang menunjukkan bahwa pengguna membutuhkan banyak langkah untuk mengontrol dan mengoperasikan sesuai karakter program yang dikembangkan.

- 4) *Integrity* merupakan karakteristik yang berkaitan dengan tingkat kontrol terhadap program oleh pengguna, baik yang mendapatkan otorisasi ataupun tidak (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Auditability*

Karakteristik yang menunjukkan kemudahan program-program untuk diuji sesuai standar.

b) *Instrumentation*

Karakteristik yang menunjukkan sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi.

c) *Security*

Karakteristik yang menunjukkan kemampuan program untuk melindungi informasi dan data. sehingga orang atau sistem yang tidak sesuai tidak dapat membaca atau memodifikasi program dan hanya orang atau sistem berwenang.

- 5) *Usability* merupakan karakteristik yang berkaitan dengan usaha yang *diperlukan* oleh pengguna untuk mengoperasikan, menyiapkan input dan menginterpretasikan output dari program (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Operability*

Karakteristik yang menunjukkan bahwa pengguna membutuhkan banyak langkah untuk mengontrol dan mengoperasikan sesuai karakter program yang dikembangkan.

b) *Training*

Karakteristik yang menunjukkan tingkat di mana program memungkinkan pengguna baru untuk menerapkan sistem.

b. *Product Revision*

- 1) Maintainability merupakan karakteristik yang berkaitan dengan usaha yang diperlukan untuk menemukan dan mengatasi kesalahan di dalam program (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

- a) *Conciseness*

Karakteristik yang menunjukkan bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang relatif ringkas.

- b) *Consistency*

Karakteristik yang menunjukkan penggunaan satu bentuk yang seragam pada seluruh rancangan sistem.

- c) *Instrumentation*

Karakteristik yang menunjukkan sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi.

- d) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

- e) *Self-documentation*

Karakteristik yang menunjukkan sejauh mana kode sumber menyediakan dokumentasi yang berarti.

f) *Simplicity*

Karakteristik yang menunjukkan tingkat, sejauh mana *user* dapat memahami program tanpa mengalami kesulitan.

- 2) *Flexibility* merupakan karakteristik yang berkaitan dengan usaha yang diperlukan untuk mengubah program yang beroperasi (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Conciseness*

Karakteristik yang menunjukkan, bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang relatif ringkas.

b) *Consistency*

Karakteristik yang menunjukkan, penggunaan satu bentuk yang seragam pada seluruh rancangan sistem.

c) *Expandability*

Karakteristik yang menunjukkan, sejauh mana tingkat standar arsitektur, data atau desain prosedural dapat dikembangkan.

d) *Generality*

Karakteristik yang menunjukkan, potensi aplikasi untuk diperluas komponennya. Generalitas didefinisikan, sejauh mana produk perangkat lunak dapat berjalan saat diperluas fungsinya.

e) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

f) *Self-documentation*

Karakteristik yang menunjukkan, sejauh mana kode sumber menyediakan dokumentasi yang berarti.

g) *Simplicity*

Karakteristik yang menunjukkan, tingkat sejauh mana *user* dapat memahami program tanpa mengalami kesulitan.

- 3) *Testability* merupakan karakteristik yang berkaitan dengan usaha yang diperlukan untuk menguji sebuah program untuk memastikan bahwa program tersebut berfungsi sebagaimana mestinya (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Audiability*

Karakteristik yang menunjukkan kemudahan program program untuk diuji sesuai standar.

b) *Instrumentation*

Karakteristik yang menunjukkan, sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi.

c) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

d) *Self-documentation*

Karakteristik yang menunjukkan, sejauh mana kode sumber menyediakan dokumentasi yang berarti.

e) *Simplicity*

Karakteristik yang menunjukkan, tingkat sejauh mana *user* dapat memahami program tanpa mengalami kesulitan.

c. ***Product Transition***

1) *Portability* merupakan karakteristik yang berkaitan dengan usaha yang diperlukan untuk mentransfer program dari satu perangkat keras atau perangkat lunak ke dalam lingkungan sistem yang berbeda (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Generality*

Karakteristik yang menunjukkan potensi aplikasi untuk diperluas komponennya. Generalitas didefinisikan sebagai sejauh mana produk perangkat lunak dapat berjalan saat diperluas fungsinya.

b) *Hardware Independence*

Karakteristik yang menunjukkan tingkat kemampuan program atau perangkat lunak untuk dipisahkan dari *hardware*.

c) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

d) *Self-documentation*

Karakteristik yang menunjukkan sejauh mana kode sumber menyediakan dokumentasi yang berarti.

e) *Software system independence*

Karakteristik yang menunjukkan tingkat di mana program ini bersifat independen (jauh dari bahasa pemrograman yang tidak standar) dan tidak mengalami kendala pada lingkungan sistem operasi.

- 2) *Reusability* merupakan karakteristik yang berkaitan dengan bagaimana sebuah program dapat digunakan kembali di dalam program lain (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Generality*

Karakteristik yang menunjukan potensi aplikasi untuk diperluas komponennya. Generalitas didefinisikan sebagai sejauh mana produk perangkat lunak dapat berjalan saat diperluas fungsinya.

b) *Hardware Independence*

Karakteristik yang menunjukan tingkat kemampuan program atau perangkat lunak untuk dipisahkan dari *hardware*.

c) *Modularity*

Karakteristik yang menunjukan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

d) *Self-documentation*

Karakteristik yang menunjukan sejauh mana kode sumber menyediakan dokumentasi yang berarti.

e) *Software system independence*

Karakteristik yang menunjukkan tingkat di mana program ini bersifat independen (jauh dari bahasa pemrograman yang tidak standar) dan tidak mengalami kendala pada lingkungan sistem operasi.

- 3) *Interoperability* merupakan karakteristik yang berkaitan dengan usaha yang diperlukan untuk menghubungkan sebuah sistem dengan sistem yang lain (Pressman, 2001: 510). Menurut Roger S. Pressman menyebutkan subkarakteristik dalam buku *Software Engineering: A Practitioner's Approach* sebagai berikut:

a) *Communication commonality*

Karakteristik yang menunjukkan bahwa tingkat standar antarmuka, *protocol (domain)* dan kapasitas *bandwith* yang digunakan untuk pengembangan aplikasi.

b) *Data Commonality*

Karakteristik yang menunjukkan tingkat penggunaan struktur data maupun jenis data yang digunakan telah memenuhi standar pengembangan program.

c) *Generality*

Karakteristik yang menunjukkan potensi aplikasi untuk diperluas komponennya. Generalitas didefinisikan sebagai sejauh mana produk perangkat lunak dapat berjalan saat diperluas fungsinya.

d) *Modularity*

Karakteristik yang menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.

17. Pengujian Software

Pengujian perangkat lunak adalah proses untuk menjalankan sebuah program atau sistem untuk mencari kesalahan. Pentingnya pengujian perangkat lunak dan implikasinya mengacu pada kualitas perangkat lunak yang tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi dengan peluang terjadi kesalahan manusia yang sangat besar dan ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna. Oleh karena itu, pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas. Proses pengujian memiliki kecenderungan untuk memandang perangkat lunak dari sisi kebutuhan pengguna. Sehingga takaran yang diukur merupakan kesesuaian dan kemampuan perangkat lunak untuk memenuhi kebutuhan pengguna akhir. Mengacu standar kualitas perangkat lunak yang dijelaskan oleh McCall, peneliti mengacu pada 10 karakteristik sebagai pengujian kualitas perangkat lunak. Berikut karakteristik sesuai produk ilmu yang dijelaskan oleh McCall dalam Pressman (2000):

a. *Product Operations*

1) Aspek *correctness*

“Correctness is the degree to which the software performs its required function. The most common measure for correctness is defects per KLOC, where a defect is defined as a verified lack of conformance to requirements” (Pressman, 2010: 680).

Pengembangan perangkat lunak aplikasi Simulasi Tes pada aspek *correctness* menggunakan mekanisme pengujian *white box*, dengan meninjau dari segi internal program. McCall (1977) menjelaskan untuk mengukur kualitas kebenaran program parameter yang paling umum adalah menggunakan cacat per *KLOC*, di mana cacat didefinisikan sebagai kurangnya verifikasi kesesuaian dengan persyaratan. Aspek *correctness* diukur dengan melakukan analisis

jumlah baris dengan teknik *Kilo Line of Code* (KLOC). Roger S. Pressman (2010: 709) menjelaskan serangkaian metode analisis *KLOC* sebagai berikut:

Tabel 2. Metode Perhitungan *Correctness* (Pressman, 2010: 709)

Metode	Rumus
Walston-Felix Model	$E = 5.2 \times (KLOC)^{0.91}$
Bailey-Basili Model	$E = 5.5 + 0.73 \times (KLOC)^{1.16}$
Boehm simple Model	$E = 3.2 \times (KLOC)^{1.05}$
Doty model for KLOC > 9 (untuk lebih dari 9000 baris kode)	$E = 5.288 \times (KLOC)^{1.047}$

Steve McConnell (2004: 698) menjelaskan ekspektasi *error* yang terjadi dalam proses pengembangan perangkat lunak berdasarkan ukuran proyek perangkat lunak dapat dilihat pada tabel 3 berikut:

Tabel 3. Ukuran Proyek dan Densitas *Error* (McConnel, 2004: 698)

Ukuran Proyek	Densitas Error
Kurang dari 2K	0-25 <i>Error per KLOC</i>
2K-16K	0-40 <i>Error per KLOC</i>
16K-64K	0.5 – 50 <i>Error per KLOC</i>
64K-512K	2-7 <i>Error per KLOC</i>
Lebih dari 512K	4-100 <i>Error per KLOC</i>

2) Aspek *Reliability*

McCall mendefinisikan *reliability* sebagai tingkat kesesuaian perangkat lunak terhadap aspek *accuracy*, *consistency*, *error tolerance*, *modularity*, *simplicity*. Sub-karakteristik aspek *reliability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 4. Sub-karakteristik Aspek *Reliability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Accuracy</i>	"Apakah sistem berjalan secara presisi?"
<i>Consistency</i>	"Apakah sistem dibangun dengan desain yang konsisten?"
<i>Error tolerance</i>	"Dapatkah sistem menunjukan kesalahan proses?"
<i>Modularity</i>	"Apakah sistem dibangun terintegrasi secara modular?"
<i>Simplicity</i>	"apakah sistem terintegrasi dengan kesederhanaan antarmuka yang memudahkan pengguna?"

"Evaluation of a software metrics confirms or rejects a correlation between internal software attributes assessed by software metrics and other external

quality attributes. Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way" (Lincke, R, 2007: 35). Dijelaskan oleh Rüdiger Lincke evaluasi metrik perangkat lunak dilakukan dengan mengevaluasi kontrol, kehandalan secara berulang oleh pihak ketiga dalam hal ini adalah pakar rekayasa perangkat lunak.

Pengujian aspek *reliability* atau kehandalan melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan *menggunakan* metode kuisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

3) Aspek *Efficiency*

McCall mendefinisikan aspek *efficiency* sebagai tingkat kesesuaian terhadap aspek *conciseness*, *execution efficiency*, *operability*. Sub-karakteristik aspek *efficiency* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 5. Sub-karakteristik Aspek *Efficiency* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Conciseness</i>	"Apakah sistem dibangun dengan jumlah baris kode yang sederhana?"
<i>Execution efficiency</i>	"Apakah eksekusi program berjalan secara efektif?"
<i>Operability</i>	"Apakah sistem membutuhkan banyak langkah untuk dijalankan?"

"Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification" (Lincke, 2007: 21). Rüdiger Lincke menjelaskan dalam buku *Validation of a Standard- and Metric-Based Software Quality Model* bahwa memerlukan proses validasi untuk menegaskan koreksi implementasi sesuai

metrik perangkat lunak sesuai dengan spesifikasinya dan nilai-nilai metrik dihitung sesuai dengan yang diharapkan dengan spesifikasi metrik.

Pengujian aspek *efficiency* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

4) Aspek *Integrity*

McCall mendefinisikan aspek *integrity* sebagai tingkat kesesuaian terhadap aspek *auditability*, *instrumentation* dan *security*. Sub-karakteristik aspek *integrity* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 6. Sub-karakteristik Aspek *Integrity* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Auditability</i>	"Apakah sistem dibangun dengan jumlah baris kode yang sederhana?"
<i>Instrumentation</i>	"Apakah eksekusi program berjalan secara efektif?"
<i>Security</i>	"Dapatkah sistem mencegah akses yang tidak mendapatkan otorisasi?"

"*Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification*" (Lincke, 2007: 21). Rüdiger Lincke menjelaskan dalam buku *Validation of a Standard- and Metric-Based Software Quality Model* bahwa memerlukan proses validasi untuk menegaskan koreksi implementasi sesuai metrik perangkat lunak sesuai dengan spesifikasinya dan nilai-nilai metrik dihitung sesuai dengan yang diharapkan dengan spesifikasi metrik.

Pengujian aspek *integrity* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisioner oleh pakar rekayasa perangkat lunak yang telah

divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

5) Aspek *Usability*

McCall mendefinisikan aspek *usability* sebagai tingkat kesesuaian terhadap aspek *operability* dan *training*. Sub-karakteristik aspek *usability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 7. Sub-karakteristik Aspek *Usability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Operability</i>	"Apakah sistem membutuhkan banyak langkah untuk dijalankan?"
<i>Training</i>	"Dapatkah sistem membantu pengguna baru untuk menerapkan sistem?"

"Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification" (Lincke, 2007: 21).

Pengujian aspek *usability* melalui mekanisme pengujian dilakukan oleh user baru dan menggunakan angket *Computer System Usability Questionnaire* yang dipublikasikan oleh J.R Lewis dalam *International Journal of Human-Computer Interaction* pada tahun 1995.

b. *Product Revision*

1) Aspek *Maintanability*

McCall mendefinisikan aspek *maintainability* sebagai tingkat kesesuaian terhadap aspek *conciseness*, *consistency*, *instrumentation*, *modularity*, *self-documentation* dan *simplicity*. Sub-karakteristik aspek *maintanability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 8. Sub-karakteristik Aspek *Maintanability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Conciseness</i>	“Apakah sistem dibangun dengan jumlah baris kode yang sederhana?”
<i>Consistency</i>	“Apakah sistem dibangun dengan desain yang konsisten?”
<i>Instrumentation</i>	“Apakah eksekusi program berjalan secara efektif?”
<i>Modularity</i>	“Apakah sistem dibangun terintegrasi secara modular?”
<i>Self-documentation</i>	“Apakah sistem dikembangkan dengan dokumentasi kode sumber yang terstruktur?”
<i>Simplicity</i>	“apakah sistem terintegrasi dengan kesederhanaan antramuka yang memudahkan pengguna?”

“*Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification*” (Lincke, 2007: 21).

Pengujian aspek *testability* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

2) Aspek *Flexibility*

McCall mendefinisikan aspek *flexibility* sebagai tingkat kesesuaian terhadap aspek *conciseness*, *consistency*, *expandability*, *generality* *modularity*, *self-documentation* dan *simplicity*. Sub-karakteristik aspek *flexibility* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 9. Sub-karakteristik Aspek *Flexibility* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Conciseness</i>	“Apakah sistem dibangun dengan jumlah baris kode yang sederhana?”
<i>Consistency</i>	“Apakah sistem dibangun dengan desain yang konsisten?”
<i>Expandability</i>	“Apakah sistem dapat dikembangkan dengan data yang lebih kompleks?”
<i>Generality</i>	“Apakah sistem dikembangkan dengan banyak fungsi?”
<i>Modularity</i>	“Apakah sistem dibangun terintegrasi secara modular?”
<i>Self-documentation</i>	“Apakah sistem dikembangkan dengan dokumentasi kode sumber yang terstruktur?”
<i>Simplicity</i>	“apakah sistem terintegrasi dengan kesederhanaan antramuka yang memudahkan pengguna?”

“Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification” (Lincke, 2007: 21).

Pengujian aspek *flexibility* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau metric fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

3) Aspek *Testability*

McCall mendefinisikan aspek *testability* sebagai tingkat kesesuaian terhadap aspek *audibility*, *instrumentation*, *modularity*, *self-documentation* dan *simplicity*. Sub-karakteristik aspek *usability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 10. Sub-karakteristik Aspek *Testability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Auditability</i>	“Apakah sistem dibangun dengan jumlah baris kode yang sederhana?”
<i>Instrumentation</i>	“Apakah eksekusi program berjalan secara efektif?”
<i>Modularity</i>	“Apakah sistem dibangun terintegrasi secara modular?”
<i>Self-documentation</i>	“Apakah sistem dikembangkan dengan dokumentasi kode sumber yang terstruktur?”
<i>Simplicity</i>	“apakah sistem terintegrasi dengan kesederhanaan antarmuka yang memudahkan pengguna?”

“Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification” (Lincke, 2007: 21).

Pengujian aspek *testability* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

c. *Product Transition*

1) Aspek *Reusability*

McCall mendefinisikan aspek *testability* sebagai tingkat kesesuaian terhadap aspek *generality*, *hardware independence*, *modularity*, *self-documentation* dan *software system independence*. Sub-karakteristik aspek *reusability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 11. Sub-karakteristik Aspek *Reusability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Generality</i>	"Apakah sistem dikembangkan dengan banyak fungsi?"
<i>Hardware Independence</i>	"Apakah sistem dapat dibangun dengan konfigurasi yang berbeda?"
<i>Modularity</i>	"Apakah sistem dibangun terintegrasi secara modular?"
<i>Self-documentation</i>	"Apakah sistem dikembangkan dengan dokumentasi kode sumber yang terstruktur?"
<i>Software System Independence</i>	"Apakah aplikasi dikembangkan sesuai dengan bahasa pemrograman yang mendukung?"

"*Validation of a software metric confirms or rejects the correctness of a given implementation of that particular software metric regarding its specification. That is, the calculated metric values correspond to values which are expected by the metric specification*" (Lincke, 2007: 21).

Pengujian aspek *reusability* melalui mekanisme pengujian *Black Box* untuk setiap komponen atau *metric* fungsional yang terdapat dalam aplikasi dengan menggunakan metode kuisioner oleh pakar rekayasa perangkat lunak yang telah divalidasi oleh Prof. Herman Dwi Surjono, Ph.D, Handaru Jati, ST, M.M, Ph.D dan Rahmatul Irfan, S.T, MT.

2) Aspek *Interoperability*

McCall mendefinisikan aspek *interoperability* sebagai tingkat kesesuaian terhadap aspek *communication commonality*, *data commonality*, *generality*, dan *modularity*. Sub-karakteristik aspek *interoperability* tersebut dijelaskan pada tabel sebagai berikut:

Tabel 12. Sub-karakteristik Aspek *Interoperability* (Pressman, 2000)

Subkarakteristik	Penjelasan
<i>Communication Commonality</i>	"Apakah sistem dibangun menggunakan antarmuka yang standar, protocol dan <i>bandwidth</i> ?"
<i>Data Commonality</i>	"Apakah sistem dibangun dengan komponen <i>database</i> yang sesuai dalam penggunaanya?"
<i>Generality</i>	"Apakah sistem dikembangkan dengan banyak fungsi?"
<i>Modularity</i>	"Apakah sistem dibangun terintegrasi secara modular?"

B. Kajian Penelitian Yang Relevan

1. Skripsi, Muliana Sukmawati, 2009, Rancang Bangun Aplikasi Simulasi TOEFL Berbasis Android

(Sukmawati, 2009) dalam penelitiannya menjelaskan hasil pengembangan aplikasi simulasi TOEFL berbasis Android dengan menggunakan *Eclipse* dan *SQLite* sebagai *database*. Metodologi yang digunakan yaitu *MSF (Microsoft Solutions Framework)*. *MSF* merupakan gabungan antara *waterfall* model dan *spiral* model. Sehingga tahapan proses lebih terarah, namun tetap dapat diiterasi.. pengembangan aplikasi ini bertujuan agar pengguna dapat berlatih tes TOEFL, mengerjakan soal-soal, dan meningkatkan kemampuan berbahasa Inggris. Aplikasi ini diharapkan mampu memberikan kemudahan pengguna dalam melakukan latihan tes TOEFL di mana pun dan kapan pun.

2. Skripsi, Imam Ahfas, 2012, Rancang Bangun Simulasi Tes Online Calon Pegawai Negeri Sipil (CPNS)

(Ahfas, 2012) dalam penelitiannya menjelaskan hasil pengembangan program simulasi tes berbasis web untuk penerimaan Calon Pegawai Negeri Sipil. Aplikasi Simulasi Tes yang dikembangkan bertujuan untuk membantu persiapan CPNS memasuki tes seleksi yang sebenarnya, sehingga memudahkan *user* melakukan tes, tanpa harus membeli buku ujian atau

bimbingan belajar. Aplikasi dikembangkan dengan materi yang memiliki kriteria soal ujian CPNS, yang meliputi psikotes, pemerintahan, pendidikan maupun ilmu pengetahuan umum. Sistem dirancang dengan menggunakan pemodelan UML (*Unified Modeling Language*) dengan menggunakan bahasa pemrograman *Dreamweaver 8.0* dan *Xampp*.

3. Skripsi, Gulam Hazmin, 2013, Aplikasi Simulasi Tes Penerimaan Calon Pegawai Negeri Sipil (CPNS) Pada Pengadilan Tinggi Agama Yogyakarta Menggunakan App Inventor

(Hazmin, 2013) dalam penelitiannya menjelaskan hasil pengembangan program simulasi tes untuk penerimaan Calon Pegawai Negeri Sipil dalam bentuk *puzzle block code*. Aplikasi Simulasi Tes yang dikembangkan bertujuan mempermudah untuk mempermudah para CPNS melakukan *try-out* maupun pra-test sebelum melalui prosedur yang sebenarnya. Aplikasi tersebut dikembangkan pada *smartphone* yang mendukung mobilitas dan portabilitas agar dapat digunakan secara efektif oleh *user* (pengguna).

4. Jurnal : Aditya Nugraha, 2011, Pengembangan dan Implementasi Mobile Learning Berbasis J2ME untuk Mata Pelajaran Keterampilan Komputer dan Pengelolaan Informasi

(Nugraha, 2011) Dalam penelitiannya menjelaskan hasil penilaian produk oleh siswa terhadap aplikasi *m-learning* ini menunjukkan penilaian yang baik terhadap aspek komunikasi visual, aspek perangkat lunak, dan aspek desain pembelajaran. Oleh karena itu, aplikasi *m-learning* berbasis J2ME ini layak untuk diimplementasikan dalam pelaksanaan pembelajaran.

5. Jurnal : Sutarno, 2010, Online Interaktif Multimedia

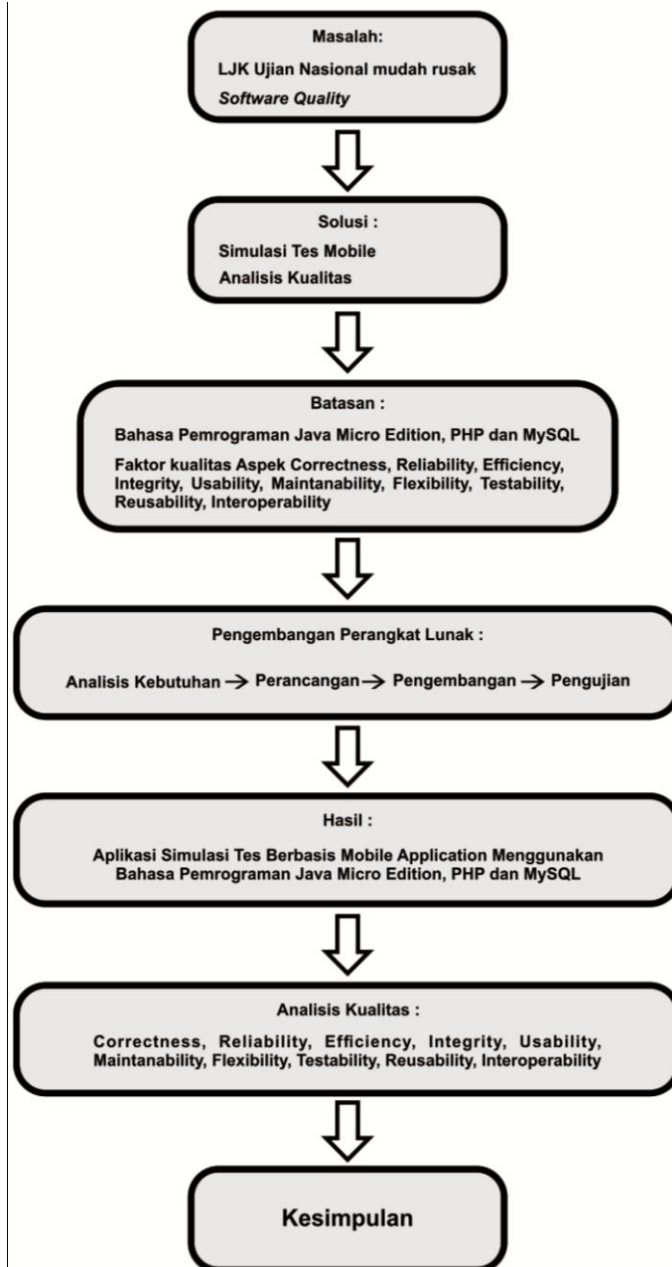
(Sutarno, 2010) Dalam penelitian yang berjudul “Online Interaktif Multimedia” memberikan penjelasan bahwa penelitian menghasilkan produk berupa aplikasi media yang berfungsi sebagai peningkatan keterampilan generik sains maupun keterampilan kritis dengan sistem evaluasi pembelajaran yang diimplementasikan ke dalam sebuah aplikasi web yang bernama Online Interaktif Multimedia. Aplikasi tersebut terdapat soal tes serupa evaluasi pembelajaran konvensional dengan kategori soal untuk meningkatkan keterampilan generik sains dan keterampilan kritis.

C. Kerangka Berfikir

Aplikasi Simulasi Tes dimaksudkan untuk memenuhi kebutuhan dan memberikan inovasi dalam pengembangan media pembelajaran yang dapat digunakan sebagai evaluasi pembelajaran. Sehingga secara tepat dan dapat mengurangi berbagai beban permasalahan yang dihadapi ketika dilakukan dengan metode evaluasi konvensional. Aplikasi dibangun dengan tahap pengembangan suatu aplikasi. Tahap-tahap tersebut meliputi tahap analisis, desain, pengembangan dan pengujian.

Dalam tahap analisis dilakukan pengumpulan informasi yang dibutuhkan untuk mengembangkan aplikasi Simulasi Tes. Tahap analisis ini mencakup analisis kebutuhan, analisis *hardware* dan analisis *software*. Dengan hasil analisis tersebut aplikasi Simulasi Tes didesain. Desain aplikasi meliputi desain struktur program yang digambarkan dengan *use case diagram*, *class diagram*, *sequence diagram* dan desain *interface* (antarmuka). Desain yang telah dihasilkan kemudian diterapkan dalam tahap pengembangan yaitu melakukan

pembuatan aplikasi Simulasi Tes. Pengembangan rekayasa perangkat lunak berupa aplikasi Simulasi Tes menggunakan *software* pembuat aplikasi berbasis java dari *oracle* yaitu Netbeans 7.0.1. Dari tahap pengembangan, diperoleh hasil berupa aplikasi Simulasi Tes. Untuk mengetahui kualitas perangkat lunak, aplikasi Simulasi Tes diuji coba dalam tahap pengujian. Tahap pengujian aplikasi Simulasi Tes yang dilakukan meliputi aspek kualitas perangkat lunak berdasarkan kriteria-kriteria dalam McCall's *Software Quality* yaitu kebenaran program (*correctness*), reliabilitas (*reliability*), efisiensi (*efficiency*), integritas (*integrity*), usabilitas (*usability*), maintainabilitas (*maintanability*), fleksibilitas (*flexibility*), testabilitas (*testability*), reusabilitas (*reusabilty*) dan interoperabilitas (*interoperability*). Tahap pengujian dilakukan untuk mengevaluasi hasil implementasi yaitu berupa evaluasi terhadap kualitas perangkat lunak, kelebihan, kekurangan, kendala, dan rekomendasi untuk aplikasi yang dikembangkan.



Gambar 7. Kerangka Berpikir

D. Pertanyaan Peneliti

Berdasarkan permasalahan yang telah dirumuskan dapat diidentifikasi beberapa pertanyaan penelitian yang diharapkan dapat terjawab dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana prosedur pengembangan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* di SMK Negeri 3 Yogyakarta?
2. Bagaimana kualitas Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* yang meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability*, *interoperability*?

BAB III

METODE PENELITIAN

A. Jenis Penelitian

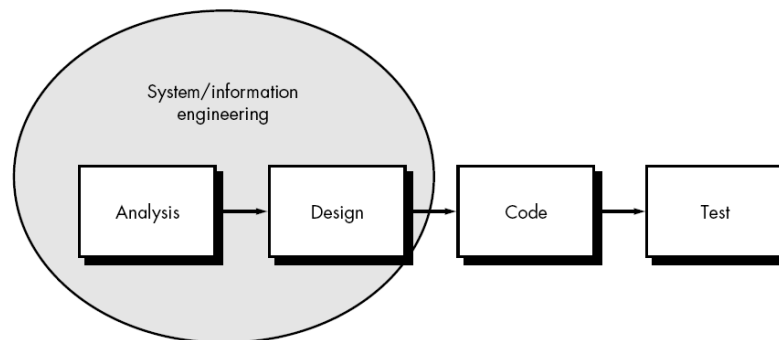
Model yang digunakan dalam penelitian “Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*” ini termasuk dalam model penelitian riset dan pengembangan (*research and development*). Pengembangan dengan model *Research and Development* menurut Borg dan Gall (1996: 782) menyebutkan “*a process used develop and validate educational product*”.

Borg and Gall memaparkan bahwasanya dalam penelitian *Research and Development (R&D)* merupakan sebuah proses untuk mengembangkan dan melakukan uji validitas produk yang dikembangkan untuk suatu kepentingan tertentu. Dalam hal ini penelitian *research and development* merupakan suatu proses dalam mengembangkan sebuah produk serta melakukan pengujian terhadap validitas produk yang dikembangkan.

Sugiyono (2009: 407) dalam bukunya menyebutkan penelitian *Research and Development* merupakan sebuah model penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektivan produk tersebut. Lebih lanjut, menurut Sugiyono penelitian R&D digunakan untuk menghasilkan sebuah produk dan diuji keefektivitasannya.

Dalam penelitian tersebut, peneliti menggunakan salah satu metode dalam membangun perangkat lunak yaitu metode *waterfall*. Metode *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun suatu

software. Langkah-langkah dalam *Waterfall Method* adalah (Pressman, 2001: 29):



Gambar 8. Waterfall Method (Pressman, 2001: 29)

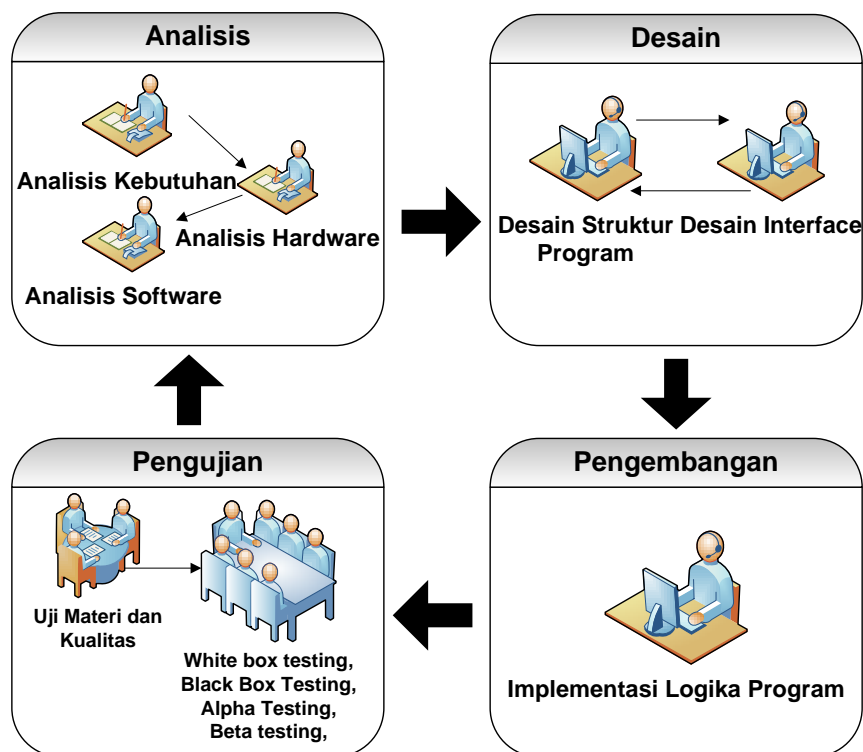
B. Desain Penelitian

Selaras dengan jenis penelitian *research and development*, proses penelitian Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* mengacu pada kaidah *software engineering* Pressman (Pressman, 2001) menjelaskan tahapan dalam pengembangan perangkat lunak, yaitu: (1) Analisis Kebutuhan, (2) Desain Sistem, (3) Pengkodean dan (4) Pengujian.

Setelah melalui tahapan pengembangan dan pengujian internal (*white box testing* dan *black box testing*), produk yang dihasilkan akan dilakukan tahap pengujian validitas. Pengujian validitas menurut Pressman meliputi pengujian *alpha* dan pengujian *beta* (Pressman, 2001). Pengujian *alpha* menurut Pressman sendiri merupakan tahapan pengujian dari sisi pengembang dengan memerhatikan aspek fundamental dari sistem dengan sedikit untuk struktur logis internal perangkat lunak. Pengujian dilakukan untuk mendapatkan *error* permasalahan penggunaan dari sisi pengembang. Pengujian *beta* merupakan pengujian pada sisi pengguna. Pengujian *beta* dilakukan untuk mendapatkan

pertimbangan perbaikan dari pengguna akhir sebelum persiapan rilis produk akhir.

Senada dengan model pengembangan pada penelitian ini, penulis mengembangkan perangkat lunak menggunakan kaidah *software engineering* model proses klasik, yaitu *waterfall's model*. Mengacu pada model tersebut beberapa hal yang akan dilakukan dalam mengembangkan produk seperti gambar berikut:



Gambar 9. Prosedur dan Alur Pengembangan

1. Analisis (*Analysis*)

Analisis merupakan tahapan di mana penulis melakukan analisa kebutuhan maupun keinginan yang diharapkan oleh pengguna. Tahapan analisa kebutuhan mencakup di dalamnya (a) analisis *hardware* (fungsional perangkat lunak) (b) desain antarmuka (*user interface*) perangkat lunak dan lain-lain. Pada tahapan

analisa kebutuhan ini diharapkan semua kebutuhan yang diharapkan pada proses pengembangan perangkat lunak dapat terpenuhi.

a. Analisis Kebutuhan

Dalam analisis kebutuhan dilakukan pengumpulan kebutuhan-kebutuhan yang digunakan dalam mengembangkan aplikasi Simulasi Tes Berbasis *Mobile Application*. Yaitu, antara lain kebutuhan sistem dan sistem yang akan dikembangkan. Kebutuhan materi mengacu pada fungsi dari subyek pengembangan.

b. Analisis *Hardware*

“Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs. Such devices can perform substantial interpretation, computation, communication, control, or other logical functions” (Meulen, 2000: 40).

Meulen mendefinisikan *hardware* sebagai perangkat yang mampu menerima dan menyimpan data komputer, mengeksekusi urutan sistematis operasi pada data komputer, atau memproduksi kontrol output. Perangkat tersebut dapat melakukan penafsiran substansial, komputasi, komunikasi, kontrol, atau fungsi Logis lainnya.

Dalam pengembangan perangkat lunak ini, jenis perangkat keras yang digunakan adalah sebagai berikut:

Tabel 13. Spesifikasi Hardware yang Digunakan

<i>Hardware</i>	<i>Komputer/Laptop</i>	<i>Handphone</i>
<i>Processor</i>	Intel Core i3	1 GHz
RAM	2.00 GB	32.00 MB
<i>Hardisk</i>	320 GB	10 MB

c. Analisis *Software*

Analisis *software* didefinisikan sebagai sekumpulan instruksi yang digunakan untuk memperoleh masukan (input) dan memanipulasinya. Sehingga menghasilkan keluaran (output) yang diinginkan dalam hal fungsi dan kinerja

yang ditentukan oleh pengguna *software* (Agarwal, Tayal, & Gupta, 2010: 4). Dalam pengembangan perangkat lunak ini, jenis perangkat lunak yang digunakan adalah:

- 1) *Windows 7 Ultimate*.
- 2) *Netbeans 7.0.1 For Windows*.
- 3) *SDK (Software Development Kit)*.
- 4) *JDK (Java Development Kit)*.
- 5) *Xampp-win 32-1.7.7-VC9*.
- 6) *SQLyog Ultimate v.10.00*.
- 7) *FileZila-3.6.02*.

2. Desain Sistem (*Design*)

Setelah melewati tahap analisa kebutuhan, maka langkah selanjutnya adalah tahapan desain sistem. Pada tahapan ini dilakukan pemodelan sistem sesuai dengan pemodelan pengembangan perangkat lunak berorientasi objek yaitu desain struktur program (*use case diagram, sequence diagram, class diagram, dan database*) dan desain *interface*. Pemodelan pada pengembangan aplikasi menggunakan *UML (Unified Modelling Language)* sebagai modelnya. Tujuan dari pemodelan sistem perangkat lunak nantinya akan mempermudah pengembang dalam proses pengkodean perangkat lunak yang dikembangkan.

3. Pengembangan

Proses pengembangan (*implementasi*) merupakan tahap mentranslasikan desain sistem ke dalam baris-baris program. Tahap pengembangan merupakan tahap membuat aplikasi Simulasi Tes Berbasis *Mobile Application* dengan menerapkan desain yang sudah dihasilkan pada tahap sebelumnya. Desain *interface* yang dirancang berdasarkan *storyboard* yang telah dihasilkan pada

tahap desain. Setelah itu, dilakukan proses pengodean dengan *software* NetBeans IDE 7.0.1. Proses ini menerjemahkan logika-logika yang sesuai dengan skenario program yang telah dirancang pada tahap desain sistem. Apabila implementasi kode program telah dilakukan, penulis melakukan uji coba apakah program dapat dijalankan dengan baik atau masih terdapat *bug*. Setelah itu, proses yang terakhir adalah melakukan *export project* ke dalam bentuk file *JAR* (Java Archive) dan *JAD* (Java Application Descriptor) sehingga dapat dijalankan pada *platform Java Mobile*.

4. Pengujian (*Evaluation*)

Tahapan akhir pada pengembangan perangkat lunak adalah pengolahan data dari hasil pengujian validasi yang dilakukan pada tahap implementasi. Pada tahap ini penyimpulan dari hasil analisis data yang diperoleh akan menunjukkan apakah produk yang dikembangkan termasuk dalam kategori layak untuk digunakan. Pada penulis juga melakukan revisi produk ketika tingkat kualitas produk tidak memenuhi standar *quality factor*.

Proses pengujian dibagi menjadi beberapa bagian sebagai berikut:

a. *White Box Testing*

“White-box testing is testing that takes into account the internal mechanism of a system or component” (IEEE, 1990). *White box testing* adalah metode pengujian yang menitikberatkan pada pengujian struktur internal program pada pengembangan perangkat lunak. Pengujian *black box* dalam pengembangannya dilakukan untuk meninjau aspek *correctness* berdasarkan analisis KLOC (*Kilo Lines Of Code*) (Pressman, 2010: 709).

b. *Black Box Testing*

“Black-box testing, also called behavioral testing, focuses on the functional requirements of the software” (Roger S Pressman, 2001: 459). *Black box testing* adalah metode pengujian yang menitikberatkan pada skenario dan fungsi sebuah perangkat lunak dengan menemukan kesalahan.

c. *Alpha testing*

Pengujian *alpha testing* dalam pengembangannya dilakukan oleh ahli rekayasa perangkat lunak. Penilaian ditinjau dari segi indikator *correctness* untuk konten program, *reliability*, *efficiency*, *integrity*, *maintainability*, *flexibility*, *testability*, *reusability* dan *interoperability* berdasarkan pedoman dari *McCall's Quality Factors*. Pengujian indikator dilakukan menggunakan *handphone* dari provider Nokia dengan emulator *Nokia For Windows* yang berbeda konfigurasi. Ahli rekayasa perangkat lunak akan memverifikasi dan memvalidasi aplikasi yang sudah dibangun. Ahli rekayasa perangkat lunak menilai unjuk kerja dari aplikasi Simulasi Tes Berbasis *Mobile Application* dan memberikan usulan terhadap aplikasi yang sudah dibangun. Tindak lanjut dari *alpha testing* ini adalah revisi tahap awal untuk perangkat lunak Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

d. *Beta Testing*

Beta testing merupakan tahap akhir pengujian penyempurnaan aplikasi Simulasi Tes yang dikembangkan. Pengujian ini dilakukan oleh pengguna akhir (*end-user*) menggunakan instrument berupa kuisisioner dari J.R. Lewis untuk mengukur aspek *usability*. Kuisisioner J.R. Lewis telah dipublikasikan dalam paper *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. Pengujian *Usability* atau pengujian tingkat kegunaan

sangat penting untuk menentukan kualitas sebuah aplikasi *mobile*. Paper tersebut telah dipublikasikan dalam *International Journal of Human Computer Interaction* pada tahun 1993. Terdapat beberapa faktor yang dapat dinilai dari tingkat kepuasan *user* ketika menggunakan sebuah aplikasi. Angket pengujian J.R. Lewis telah banyak digunakan sebagai instrumen penelitian terhadap faktor kualitas *usability*.

C. Sumber Data

1. Subjek Penelitian

Subyek penelitian untuk aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability* dan *interoperability* adalah Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*. Subyek penelitian untuk aspek *correctness*, *reliability*, *efficiency*, *integrity*, *flexibility*, *testability*, *reusability* dan *interoperability* adalah 5 pakar rekayasa perangkat lunak. Sedangkan aspek *usability* adalah 35 siswa kelas XII AV 1 Jurusan Audio Video SMK Negeri 3 Yogyakarta.

2. Sampel Penelitian

Pada proses pengujian perangkat lunak Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* penulis menggunakan sampel dalam melakukan pengembangan. Teknik *sampling* yang dipilih ialah teknik *purposive sampling*. Teknik *purposive sampling* merupakan metode sampling dimana dalam menentukan sampel didasari oleh pertimbangan-pertimbangan tertentu (Sugiyono, 2010: 124). Sampel ini digunakan untuk pengujian aspek *usability*. Jumlah sampel

menggunakan standar dari Jacob Nielsen dengan jumlah sampel 20. Jacob Neilson (2012) menjelaskan “*Test at least 20 users to get statistically significant numbers; tight confidence intervals require even more users*” (Neilson, 2012).

3. Variabel Penelitian

Variabel penelitian yang ada dalam penelitian Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL* adalah sebagai berikut:

- a. Kebenaran Program (*Correctness*)
- b. Reliabilitas (*Reliability*)
- c. Efisiensi (*Efficiency*)
- d. Integritas (*Integrity*)
- e. Usabilitas (*Usability*)
- f. Maintanabilitas (*Maintainability*)
- g. Fleksibilitas (*Flexibility*)
- h. Testabilitas (*Testability*)
- i. Reusabilitas (*Reusability*)
- j. Interoperabilitas (*Interoperability*)

4. Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada bulan Februari 2014 hingga September 2014. Pengujian aspek *reliability, efficiency, integrity, maintainability, flexibility, testability, reusability* dan aspek *interoperability* dilakukan oleh ahli rekayasa perangkat lunak dari PT. Snoov Media, PT. Sebangsa Bersama, PT. Nuwira. Sedangkan, pengujian aspek *usability* dilakukan di Laboratorium Praktik Jurusan Audio Video SMK Negeri 3 Yogyakarta.

D. Metode dan Alat Pengumpulan Data

Teknik pengumpulan data yang dipakai dalam penelitian yang berjudul Pengembangan dan Analisis Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* ini antara lain: (1) metode tes, (2) metode kuisisioner dan wawancara (Suharsimi Arikunto, 2002: 198-206). Teknik pengumpulan data dalam pengembangan ini adalah metode angket dan kuisisioner.

Kuisisioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan kepada responden untuk dijawab (Sugiyono, 2010: 199). Jenis-jenis kuisisioner dibagi dalam (Suharsimi Arikunto, 2002: 128-129):

1. Berdasarkan dari cara menjawab
 - a. Kuisisioner terbuka, yang member kesempatan kepada responden untuk menjawab dengan kalimatnya sendiri.
 - b. Kuisisioner tertutup, yang sudah disediakan jawabannya sehingga responden tinggal memilih.
2. Berdasarkan dari jawaban yang diberikan
 - a. Kuisisioner langsung, yaitu responden menjawab tentang dirinya.
 - b. Kuisisioner tidak langsung, yaitu jika responden menjawab tentang orang lain.
3. Berdasarkan dari bentuknya
 - a. Kuisisioner pilihan ganda yang dimaksud adalah sama dengan kuisisioner tertutup
 - b. Kuisisioner isian, yang dimaksud adalah kuisisioner terbuka
 - c. Check list, sebuah daftar, dimana responden tinggal membubuhkan tanda check (\checkmark) pada kolom yang sesuai

- d. Skala bertingkat, yaitu sebuah pernyataan diikuti oleh kolom-kolom yang menunjukkan tingkat-tingkatan misalnya mulai dari sangat setuju sampai sangat tidak setuju.

Alat pengumpulan data yang digunakan dalam pengujian penelitian ini antara lain:

1. Aspek *Correctness*

Pengujian aspek *correctness* terdapat 3 sub-karakteristik sesuai *McCall's Quality Factor*, yaitu: *completeness*, *consistency* dan *traceability*. Pengujian aspek tersebut menggunakan mekanisme *white box* dengan analisis perhitungan KLOC (*Kilo Line of Code*) (Pressman, 2010: 709). Hasil tersebut dihubungkan dengan ekspektasi *error* yang dikemukakan oleh Steve McConnell (2004: 698) untuk melihat cacat program internal.

2. Aspek *Reliability*

Pengujian aspek *reliability* dilakukan dalam tahap *alpha testing* yang mengandung sub-karakteristik *accuracy*, *consistency*, *error tolerance*, *modularity* dan *simplicity*. Pengujian aspek *reliability* menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

3. Aspek *Efficiency*

Pengujian aspek *efficiency* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *conciseness*, *execution efficiency* dan *operability*. Pengujian aspek *efficiency* menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

4. Aspek *Integrity*

Pengujian aspek *integrity* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *auditability*, *instrumentation* dan *security*. Pengujian aspek *integrity* terkait keamanan sebuah program dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

5. Aspek *Usability*

Pengujian aspek *usability* dilakukan pada tahap *beta testing* yang mengandung sub-karakteristik *operability* dan *training*. Pengujian aspek *usability* terkait interaksi dengan pengguna dilakukan kepada 31 responden. Instrumen *usability* menggunakan kuisioner J.R. Lewis. Tabel berikut menunjukkan instrumen aspek *usability*:

Tabel 14. *Computer System Usability Questionnaire* oleh J.R. Lewis (Lewis, 1993)

No.	Kriteria	Pernyataan
1.	<i>Operability</i>	Secara keseluruhan, saya puas dengan kemudahan pemakaian aplikasi ini.
2.		Penggunaan aplikasi ini sangat simpel/ sederhana.
3.		Saya dapat menyelesaikan pekerjaan dengan sempurna menggunakan aplikasi ini.
4.		Saya dapat menyelesaikan pekerjaan saya dengan cepat menggunakan aplikasi ini.
5.		Saya dapat menyelesaikan pekerjaan saya secara efisien menggunakan aplikasi ini.
6.		Saya merasa nyaman menggunakan aplikasi ini.
7.		Sangat mudah mempelajari penggunaan aplikasi ini.
8.		Saya yakin akan lebih produktif ketika menggunakan aplikasi ini.
9.		Pesan kesalahan yang diberikan aplikasi ini menjelaskan dengan jelas cara mengatasinya.
10.		Kapanpun saya membuat kesalahan, saya bisa memperbaikinya dengan cepat dan mudah.
11.	<i>Training</i>	Informasi yang disediakan aplikasi ini cukup jelas.
12.		Sangat mudah mencari informasi yang dibutuhkan di aplikasi ini.
13.		Informasi yang disediakan aplikasi sangat mudah dipahami.
14.		Informasi yang disediakan efektif membantu menyelesaikan tugas.
15.		Tata letak informasi yang ditampilkan di layar aplikasi sangat jelas.
16.		Tampilan/ antarmuka aplikasi menyenangkan.
17.		Saya suka menggunakan tampilan aplikasi seperti ini.
18.		Aplikasi ini memiliki fungsi dan kemampuan (kapabilitas) sesuai harapan saya.
19.		Secara keseluruhan, saya puas dengan aplikasi ini.

6. Aspek Maintainability

Pengujian aspek *maintainability* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *conciseness*, *consistency*, *instrumentation*, *modularity*, *self-documentation* dan *simplicity*. Pengujian aspek *maintainability* terkait usaha program untuk menemukan dan memperbaiki kesalahan dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

7. Aspek Flexibility

Pengujian aspek *flexibility* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *conciseness*, *consistency*, *expandability*, *generality*, *modularity*, *self-documentation* dan *simplicity*. Pengujian aspek *flexibility* menekankan pada pengujian modifikasi sebuah program dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

8. Aspek Testability

Pengujian aspek *testability* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *auditability*, *instrumentation*, *modularity*, *self-documentation* dan *simplicity*. Pengujian aspek *testability* memastikan fungsi program berjalan sesuai pada perencanaan dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

9. Aspek *Reusability*

Pengujian aspek *reusability* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *generality*, *hardware independence*, *modularity*, *self-documentation* dan *software system independence*. Pengujian aspek *reusability* menekankan pengujian bagian program (kode program) dapat digunakan kedalam lingkungan yang berbeda. Pengujian ini dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

10. Aspek *Interoperability*

Pengujian aspek *interoperability* dilakukan pada tahap *alpha testing* yang mengandung sub-karakteristik *communication commonality*, *data commonality*, *expandability* dan *modularity*. Pengujian aspek *interoperability* menekankan pada pengujian menggabungkan dengan sistem yang berbeda. Pengujian ini dilakukan menggunakan kuisioner yang telah divalidasi oleh pakar rekayasa perangkat lunak. *Software metrics are evaluated in a controlled, reliable, and repeatable (by third parties) way*" (Lincke, 2007: 35).

E. Validitas dan Reliabilitas Instrumen

1. Validitas

Instrumen dikatakan valid apabila instrumen tersebut dapat dengan tepat mengukur apa yang akan diukur (Widoyoko, 2012;141; Hasan, 2004: 15). Dengan kata lain, validitas berkaitan dengan "ketepatan" alat ukur. Dengan instrumen yang valid akan menghasilkan data yang valid pula. Validitas instrumen yang digunakan dalam penelitian ini adalah validitas konstruk

(*Construct Validity*). Validitas isi mengukur cakupan instrumen yang seharusnya diukur. Apakah instrumen yang digunakan sudah sesuai/relevan dengan apa yang akan diukur atau belum. Validasi instrumen dilakukan dengan cara menyusun kisi-kisi dan konsultan ahli (*expert judgement*). Validasi instrumen dilakukan oleh tiga pakar ahli yaitu Prof. Herman Dwi Surjono, Ph.D, Handaru Djati, Ph.D, Rahmatul Irfan, MT. Selain tiga pakar tersebut, validitas isi juga dilakukan dengan menggunakan instrumen pada 5 pakar ahli lainnya. Analisis validitas isi pada penelitian ini menggunakan perhitungan Lawshe (1975: 567). Data analisa validitas isi didapatkan dari instrumen validasi isi pakar ahli.

Suatu butir instrumen dikatakan valid apabila memiliki sumbangan besar terhadap skor total. Dengan kata lain, mempunyai validitas tinggi jika skor pada butir instrumen mempunyai kesejajaran skor total. Kesejajaran ini dapat diartikan dengan korelasi. Sehingga untuk mengetahui validitas butir digunakan rumus korelasi *product moment*. Korelasi yang digunakan pada penelitian adalah korelasi pearson dengan rumus sebagai berikut (Widoyoko, 2012: 147):

$$r_{xy} = \frac{n \sum XY - (\sum X)(\sum Y)}{\sqrt{[n \sum X^2 - (\sum X)^2][n \sum Y^2 - (\sum Y)^2]}}$$

Keterangan :

r_{xy} = koefisien korelasi Pearson variabel X dan Y

X = skor butir

Y = skor total

Data hasil analisa instrumen validitasi isi nantinya akan diolah dan dikomparasikan dengan CVR (*Content Validity Rasio*). Jika nilai $CVR_{hitung} \geq CVR_{tabel}$, maka item dinyatakan relevan. Sedangkan, jika $CVR_{hitung} < CVR_{tabel}$

maka item dinyatakan tidak relevan. CVR_{tabel} untuk 5 pakar ahli adalah sebesar 0.99. Hasil perhitungan validitas isi dapat dilihat pada tabel berikut:

Tabel 15. Hasil Perhitungan *CVR*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1	<i>Reliability</i>	14	14	14	14	14	70	0,99	Valid
2	<i>Efficiency</i>	8	8	8	8	8	40	0,99	Valid
3	<i>Integrity</i>	18	18	18	18	18	90	0,99	Valid
4	<i>Maintainability</i>	14	14	14	14	14	70	0,99	Valid
5	<i>Flexibility</i>	16	16	16	16	16	80	0,99	Valid
6	<i>Testability</i>	15	15	15	15	15	75	0,99	Valid
7	<i>Reusability</i>	12	12	12	12	12	60	0,99	Valid
8	<i>Interoperability</i>	7	7	7	7	7	35	0,99	Valid
	Total	104	104	104	104	104	520		

Validasi konstruk mengacu pada sejauh mana suatu instrumen mengukur konsep dari suatu teori yang menjadi dasar penyusunan instrumen (Widoyoko, 2012: 145). Oleh karena itu, harus ada pembahasan mengenai teori tentang variabel yang akan diukur yang menjadi dasar penentuan konstruk instrumen. Validitas konstruk didapatkan dari instrumen hasil uji coba instrumen. Data hasil uji coba kemudian dilakukan uji empirik dengan analisis korelasi (*pearson correlation*). Analisis korelasi dilakukan dengan tujuan untuk mengetahui sejauh mana butir instrumen berkorelasi dengan variabel yang diukur dalam penelitian (Westen & Rosenthal, 2003: 608). Sehingga akan diketahui sejauh mana keterkaitan antar variabel yang diukur. Validitasi butir instrumen ditentukan dengan membandingkan r_{hitung} (nilai *pearson correlation*) pada masing-masing butir dengan nilai r_{tabel} . Nilai signifikansi yang diambil adalah 0,55 dengan uji dua sisi dan $df = 35$, maka didapat nilai r_{tabel} adalah 0,325 (untuk variabel *usability*), sedangkan $df = 5$ didapat nilai r_{tabel} adalah 0,754 (untuk variabel *reliability*, *efficiency*, *integrity*, *maintainability*, *flexibility*, *testability*, *reusability*, *interoperability*). Jika nilai butir positif dan $r_{hitung} \geq r_{tabel}$, maka item dinyatakan valid. Sedangkan, jika $r_{hitung} < r_{tabel}$ maka item dinyatakan tidak valid.

a. Instrumen *Usability*

Uji validitas item pernyataan pada instrumen *usability* menggunakan kuisioner JR. Lewis dilakukan dengan bantuan program *SPSS Statistics 17 for Windows*. Hasil perhitungan menunjukkan bahwa butir pernyataan dinyatakan valid semua sehingga butir pernyataan dapat digunakan semuanya. Detail hasil perhitungan validitas item pada instrumen *usability* kuisioner JR. Lewis dilihat pada tabel sebagai berikut:

Tabel 16. Validitas Instrumen *Usability*

Butir Pernyataan	Koefisien Korelasi	Keterangan
Item 1	0,735	Valid
Item 2	0,735	Valid
Item 3	0,735	Valid
Item 4	0,653	Valid
Item 5	0,718	Valid
Item 6	0,774	Valid
Item 7	0,774	Valid
Item 8	0,653	Valid
Item 9	0,774	Valid

Dapat dilihat pada tabel di atas bahwa hasil perhitungan dengan bantuan program *SPSS Statistics 17 for Windows*. Hasil perhitungan menunjukkan bahwa butir pernyataan pada instrumen dinyatakan valid semua dan dapat digunakan sebagai pengambilan data.

b. Instrumen *Reliability*

Uji validitas instrumen *reliability* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 17. Hasil Perhitungan Validitas Instrumen *Reliability*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Reliability</i>	5	5	5	5	5	35	0.99	Valid

c. Instrumen *Efficiency*

Uji validitas instrumen *efficiency* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 18. Hasil Perhitungan Validitas Instrumen *Efficiency*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Efficiency</i>	3	3	3	3	3	15	0.99	Valid

d. Instrumen *Integrity*

Uji validitas instrumen *integrity* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 19. Hasil Perhitungan Validitas Instrumen *Integrity*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Integrity</i>	3	3	3	3	3	15	0.99	Valid

e. Instrument *Maintanability*

Uji validitas instrumen *maintanability* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 20. Hasil Perhitungan Validitas Instrumen *Maintanability*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Maintanability</i>	3	3	3	3	3	15	0.99	Valid

f. Instrumen *Flexibility*

Uji validitas instrumen *flexibiltiy* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah

0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 21. Hasil Perhitungan Validitas Instrumen *Flexibility*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Maintanability</i>	4	4	4	4	4	20	0.99	Valid

g. Instrumen *Testability*

Uji validitas instrumen *testability* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 22. Hasil Perhitungan Validitas Instrumen *Testability*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Testability</i>	4	4	4	4	4	20	0.99	Valid

h. Instrumen *Reusability*

Uji validitas instrumen *reusability* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 23. Hasil Perhitungan Validitas Instrumen *Reusability*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Testability</i>	4	4	4	4	4	20	0.99	Valid

i. Instrumen *Interoperability*

Uji validitas instrumen *interoperability* dilakukan dengan menggunakan perhitungan CVR (*Content Validity Rasio*). Nilai signifikansi yang diambil adalah 0,99 dengan nilai maksimal dengan batas nilai minimal 0.05. Dapat dilihat secara detail hasil perhitungan CVR pada tabel berikut:

Tabel 24. Hasil Perhitungan Validitas Instrumen *Interoperability*

No.	Item	A	B	C	D	E	Total	CVR	Ket
1.	<i>Interoperability</i>	4	4	4	4	4	20	0.99	Valid

2. Reliabilitas

Kata reliabilitas dalam bahasa Indonesia diambil dari kata *reliability* dalam bahasa Inggris. Reliabilitas artinya memiliki sifat dapat dipercaya (Hasan, 2004: 15). Suatu alat ukur dikatakan memiliki reliabilitas apabila dipergunakan berkali-kali oleh peneliti yang sama atau oleh peneliti yang lain tetap memberikan hasil yang sama. Jadi, reliabilitas adalah seberapa jauh konsistensi alat ukur untuk dapat memberikan hasil yang sama dalam mengukur hal dan subyek yang sama. Iqbal Hasan (2004: 15) mengemukakan bahwa reliabilitas mengandung tiga makna, yaitu tidak berubah-ubah, konsisten, dan dapat diandalkan.

Untuk mengetahui apakah instrumen tersebut reliabel atau tidak dapat dikonsultasikan dengan harga kritik atau standar realibilitas. Harga standar reliabilitas instrumen adalah 0,7 (Widoyoko, 2012: 165). Artinya, suatu instrumen dikatakan reliabel jika mempunyai nilai koefisien *Alpha* sekurang-kurangnya adalah 0,7. Untuk mengukur reabilitas atau mencari nilai alpha dapat ditunjukan pada tabel berikut:

Tabel 25. Reliabilitas Instrumen

Responden Ahli	RE	EF	IN	MN	FL	TS	RS	IT	Total	CVR	Ket
A	7	3	3	3	4	4	4	4	32	0,99	Reliabel
B	7	3	3	3	4	4	4	4	32	0,99	Reliabel
C	7	3	3	3	4	4	4	4	32	0,99	Reliabel
D	7	3	3	3	4	4	4	4	32	0,99	Reliabel
E	7	3	3	3	4	4	4	4	32	0,99	Reliabel

Keterangan :

RE : *Reliability*
 EF : *Efficiency*
 IN : *Integrity*

MN : *Maintanability*
FL : *Flexibility*
TS : *Testability*
RS : *Reusability*
IT : *Interoperability*

F. Teknik Analisis Data

Jenis data yang diperoleh dalam pengembangan ini adalah jenis data kuantitatif. Setelah dilakukan pengambilan data kemudian dilakukan analisis data. Analisa data yang digunakan pada penelitian ini menggunakan teknik analisa skala Likert. Analisa dengan pendekatan ini sesuai dengan pengukuran yang digunakan pada angket yaitu skala Guttman dan Likert. Proses analisa ini digunakan untuk menghitung data variabel yang diujikan yaitu *McCall's Quality Factor* pada aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintainability*, *flexibility*, *testability*, *reusability* dan *interoperability*.

1. Analisis White Box Testing

Proses analisis pada aspek *correctness* dilakukan menggunakan metode pendekatan perhitungan jumlah *error* (E) per *Kilo Line of Code* (KLOC) menurut model Waltson-Felix, Bailey-Basili, Boehm, dan Doty (apabila ukuran KLOC > 9). Jika jumlah *error* yang diperoleh dari kalkulasi KLOC berada pada rentang nilai *error* pada *Densitas Error* yang sesuai, dapat disimpulkan bahwa perangkat lunak yang diuji memenuhi *software quality* dari aspek *correctness*.

Metode perhitungan *Kilo Line of Code* (KLOC) menurut Waltson-Felix, Bailey-Basili, Boehm, dan Doty adalah sebagai berikut:

Tabel 26. Ukuran Proyek dan Densitas *Error* (Pressman, 2010: 709)

Ukuran Proyek	Densitas Error
Kurang dari 2K	0-25 Error per KLOC
2K-16K	0-40 Error per KLOC
16K-64K	0.5 – 50 Error per KLOC
64K-512K	2-7 Error per KLOC
Lebih dari 512K	4-101 Error per KLOC

Standar yang digunakan sebagai acuan adalah standar *defects per KLOC* bahasa pemrograman Java yakni 61 *defects per KLOC*. Apabila aplikasi yang diuji mampu memenuhi standar pengujian 100% *correctness* berjalan dengan baik dan nilai yang diperoleh kurang dari 61 (< 61) dapat disimpulkan bahwa aplikasi yang diuji tersebut telah memenuhi aspek *software quality* ditinjau dari aspek *correctness*.

2. Analisis *Black Box Testing*

Proses analisis pada pengujian *black box* dengan mekanisme pengujian fungsi perangkat lunak dengan teknik memberikan kondisi (input) sesuai dengan alur dan reaksi perangkat lunak yang dikembangkan. Pendekatan yang dilakukan adalah meninjau kesalahan dengan kategori: (1) terdapat fungsi yang hilang, (2) kesalahan antarmuka, (3) kesalahan dalam struktur data dan eksternal *database*, (4) perilaku atau kinerja kesalahan, (5) inisialisasi dan kesalahan terminasi. (Pressman, 2010: 495)

3. Analisis *Alpha Testing*

Proses analisis pada aspek *reliability*, *efficiency*, *integrity*, *maintainability*, *flexibility*, *testability*, *reusability* dan *interoperability* dilakukan menggunakan analisis Likert. Berdasarkan hasil analisis instrumen nantinya akan didapatkan skor per instrumen kemudian nanti akan dihitung rata-rata dari instrument dengan menggunakan rumus:

(1) Rumus Perhitungan rata-rata instrumen :

$$\bar{x} = \frac{\sum x}{n}$$

Dimana : \bar{x} = Skor rata-rata
 $\sum x$ = Skor total item
n = Jumlah Item

(2) Rumus perhitungan persentase skor ditulis dengan rumus berikut :

$$\text{Persentase Kelayakan}(\%) = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

Setelah didapatkan hasil presentasi dari perhitungan sebelumnya, kemudian data dikonversi ke dalam pernyataan predikat. Untuk melakukan konversi dari hasil perhitungan data ke dalam pernyataan predikat digunakan skala Likert. Konversi persentase ke pernyataan seperti dalam tabel seperti berikut (Riduwan, 2011: 88; Aryoyudanta, 2012):

Tabel 27. Interpretasi Persentase Likert

No	Presentase	Interpretasi
1	0% - 20%	Sangat Tidak Baik
2	21% - 40%	Tidak Baik
3	41% - 60%	Cukup Baik
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

Supaya konversi persentase ke dalam bentuk pernyataan lebih sesuai dengan penelitian yang dilakukan, maka skala konversi persentase di atas disesuaikan interpretasinya. Dari hasil perhitungan analisa data penelitian nantinya akan didapat interpretasi kelayakan perangkat lunak yang dikembangkan dan diteliti. Hasil penelitian ini nantinya akan menentukan kualitas perangkat lunak baik per-faktor kualitas maupun secara keseluruhan.

4. Analisis Beta Testing

Analisis *usability* dilakukan dengan menguji nilai konsistensi internal data yang diperoleh melalui penelitian menggunakan kuisioner internal data yang diperoleh melalui penelitian menggunakan kuisioner *usability* J.R Lewis dengan menggunakan metode Alpha Cronbach.

$$\alpha = \frac{K}{K-1} \left(1 - \frac{\sum_{i=1}^K \sigma_{Y_i}^2}{\sigma_X^2} \right)$$

Nilai yang diperoleh dalam perhitungan tersebut kemudian dikomparasikan dengan tabel nilai konsistensi Alpha Cronbach. Semakin tinggi nilai konsistensi internal yang diperoleh dalam penelitian, nilai aspek *usability* produk semakin baik dan produk yang diuji layak digunakan oleh pengguna.

Tabel 28. Nilai Konsistensi Internal Alpha Cronbach

Cronbach's Alpha	Internal Consistency
$\alpha \geq .9$	Sangat Baik
$.9 > \alpha \geq .8$	Baik
$.8 > \alpha \geq .7$	Dapat Diterima
$.7 > \alpha \geq .6$	Diragukan
$.6 > \alpha \geq .5$	Buruk
$.5 > \alpha$	Tidak dapat diterima

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Pengembangan Perangkat Lunak

1. Analisis Kebutuhan Sistem

Dalam tahap analisis dilakukan pengumpulan informasi yang dibutuhkan untuk mengembangkan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*. Tahap analisis ini mencakup analisis kebutuhan, analisis *hardware*, dan analisis *software*.

Berdasarkan hasil observasi evaluasi pembelajaran di SMK N 3 Yogyakarta dengan produk yang dikembangkan, aplikasi dikembangkan untuk 2 kategori *user* yaitu: guru dan siswa. Berikut adalah sasaran pengembangan aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* ini antara lain:

- a. Aplikasi Simulasi Tes dikembangkan dengan *platform mobile* khusus sebagai bagian utama diperuntukan bagi siswa.
- b. Aplikasi Simulasi Tes dikembangkan dengan *Web Service* sebagai monitoring hasil evaluasi siswa diperuntukan untuk *administrator* dan guru.

Sistem yang dikembangkan membutuhkan sebuah *server* guna menampung database berupa data *user* maupun materi soal sebagai konten utama.

2. Analisis Kebutuhan Fitur

Analisis fitur aplikasi dikelompokkan berdasarkan kategori *user* yang menjalankan skenario aplikasi:

a. Sistem Guru/Admin:

- 1) Guru/admin dapat melakukan manajemen *client/user* dengan baik terkait hak akses untuk simulasi tes.
- 2) Guru/admin dapat manajemen nomor ujian sesuai dengan *client/user*.
- 3) Guru/admin dapat memonitor seluruh nilai *client/user* setelah mengerjakan simulasi tes.
- 4) Guru dapat manajemen konten soal yang disajikan untuk simulasi tes.

b. Sistem *User/Siswa*

- 1) *Client/user* dapat login aplikasi simulasi tes *mobile*.
- 2) *Client/user* dapat memilih mata pelajaran yang diujikan dalam aplikasi.
- 3) *Client/user* dapat *mengerjakan* soal yang disajikan dalam aplikasi.
- 4) *Client/user* dapat mengunggah nilai setelah mengerjakan soal dalam aplikasi.
- 5) *Client/user* dapat melihat hasil nilai sesuai id *user* pada aplikasi web.
- 6) *Client/user* dapat mengakses hasil nilai.
- 7) *Client/user* dapat mengakses hasil pembahasan soal Simulasi Tes.
- 8) *Client/user* dapat mengerjakan soal pengayaan dalam aplikasi Simulasi Tes. Jika, nilai tidak memenuhi standar nilai dalam aplikasi Simulasi Tes *mobile*.

c. Ketentuan Penggunaan Aplikasi/Sistem

- 1) Admin/*user* telah terdaftar dalam *database* sistem atau *autorisation user* (nama, id *user*, *password*, alamat, tipe *user*).

- 2) Admin/user telah melakukan pemasangan aplikasi latihan tes ke dalam *handphone* bersistem operasi Java.
- 3) Admin/user melakukan *login* menggunakan id *user* dan password yang telah terdaftar sebelum mengerjakan soal dalam aplikasi.
- 4) Admin/user dapat mengakses aplikasi web simulasi tes dengan menuju alamat <https://rizam.fastacenter.com>.
- 5) Admin/user melakukan *login* aplikasi web simulasi tes, setelah mengerjakan soal pada aplikasi latihan tes *mobile*.

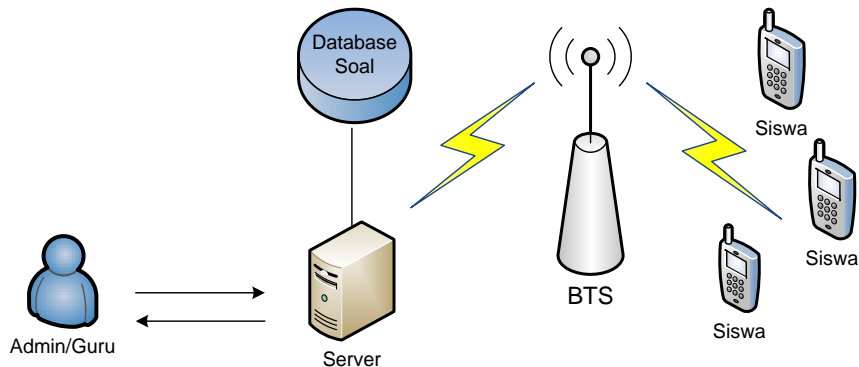
3. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Agar bisa mencapai *user experience* yang ideal, kebutuhan perangkat keras maupun lunak yang diperlukan untuk menjalankan Aplikasi Simulasi Tes Berbasis *Mobile Application* ini adalah:

- a. *Handphone* memiliki sistem operasi *Java* ataupun *Symbian OS (Operating System)* dengan fitur *Java OS (Operating System)* dengan versi minimal 1.0 dan prosesor minimal 300MHz.
- b. *SDK (Software Development Kit)* dari *Nokia* versi 1.1, 2.0 sebagai emulator aplikasi.
- c. Ukuran layar 3.5 inchi dengan resolusi minimal 96x54 piksel.
- d. *Web browser* untuk mengakses aplikasi *web* simulasi tes yaitu: *Mozilla, Google Chrome, Explorer, Opera*
- e. Perangkat komputer dengan ketersediaan koneksi internet.

B. Tahapan Desain

1. Desain Sistem



Gambar 10. Gambaran Umum Aplikasi Simulasi Tes

Desain perangkat lunak adalah gambaran aktivitas yang difokuskan pada spesifikasi detail dari solusi berbasis komputer. Pada gambar di atas menjelaskan desain sistem aplikasi Simulasi Tes berbasis *mobile application* secara umum (*general system design*) yang akan disampaikan kepada *stakeholder* sistem dan spesifikasi desain rinci yang akan digunakan pada tahap implementasi. Desain sistem aplikasi Simulasi Tes dirancang bertujuan untuk memenuhi kebutuhan kepada pemakai sistem, selain itu juga memberikan gambaran yang jelas yang lengkap kepada program komputer dan ahli-ahli teknik lainnya yang terlibat.

2. Perancangan *Unified Modeling Language*

Unified Modelling Language (UML) berguna untuk menggambarkan rancangan keseluruhan sistem. Diagram UML yang digunakan dalam penelitian ini sebagai representasi alur kerja dari sistem adalah: *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram*.

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk membentuk tingkah laku benda/*things* direalisasikan oleh *collaboration*. *Use case diagram* menjelaskan

sebuah aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Secara umum dalam pengembangan aplikasi Simulasi Tes *mobile*, *use case diagram* digunakan untuk mengetahui fungsi yang ada di dalam sebuah sistem dan menjelaskan peranan aktor yang terhadap aplikasi yang dikembangkan. *Use case diagram* untuk aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* ini menjelaskan skenario guru dan siswa sebagai aktor dalam setiap kasus penggunaan aplikasi untuk melakukan aktivitas pada sistem.

Berikut adalah penjelasan mengenai aktivitas aktor berdasarkan *use case diagram* pada aplikasi:

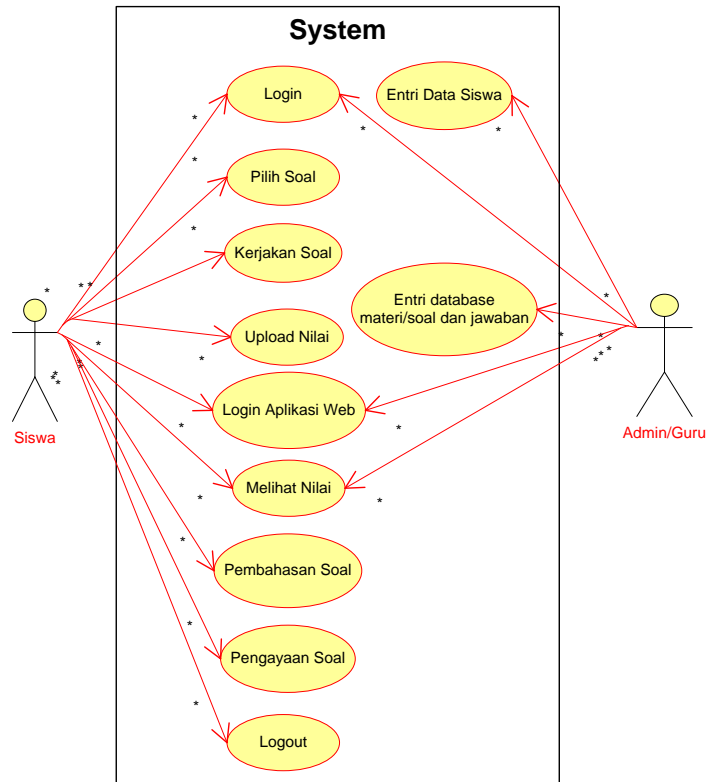
1) Aktivitas Admin/Guru

- a) Admin/Guru melakukan entri data siswa yang berhak melakukan ujian Simulasi Tes.
- b) Admin/Guru melakukan entri data materi/soal yang diujikan dan jawaban dalam aplikasi Simulasi Tes.
- c) Admin/Guru dapat melakukan login pada aplikasi Simulasi Tes *mobile*.
- d) Admin/Guru dapat melakukan login pada aplikasi Web Simulasi Tes.
- e) Admin/Guru dapat melakukan monitoring seluruh nilai siswa melalui aplikasi web Simulasi Tes

2) Aktivitas Siswa

- a) Siswa melakukan *login* pada aplikasi Simulasi Tes *mobile* setelah terdaftar sebagai *user* di dalam sistem.
- b) Siswa memilih soal pada halaman “Menu” sesuai mata pelajaran yang disajikan dalam aplikasi *mobile*.

- c) Siswa mengerjakan soal sesuai mata pelajaran yang dipilih setelah masuk halaman “Test” .
- d) Setelah selesai mengerjakan soal, siswa mendapatkan pesan untuk memproses jawaban yang dipilih dan mengunggah nilai pada halaman “Upload”.
- e) Setelah selesai mengerjakan seluruh soal yang disajikan pada aplikasi Simulasi Tes mobile, siswa melakukan login pada aplikasi web Simulasi Tes.
- f) Siswa dapat melihat hasil dalam bentuk nilai pada halaman “Nilai” aplikasi web Simulasi Tes.
- g) Siswa dapat melihat halaman “Pembahasan” saat soal pada aplikasi web Simulasi Tes.
- h) Siswa melakukan pengayaan nilai hasil evaluasi jika nilai kurang mencukupi standar dari guru dengan mengerjakan soal pada halaman “Pengayaan” aplikasi Web Simulasi Tes.



Gambar 11. Rancangan *Use Case Diagram*

Untuk skema interaksi antara pengguna dengan sistem dipaparkan dalam skenario pada tabel *use case* sebagai berikut:

Tabel 29. *Use Case* Proses Memulai Aplikasi

Use case ID	ST-001
Use case Name	Memulai Aplikasi
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	User/Siswa
Description	Use case ini melakukan proses menjalankan aplikasi.
Pre-conditions	Halaman <i>Splash Screen</i>
Basic Path	1. <i>Client/user</i> membuka aplikasi. 2. <i>Splash screen</i> mengawali sistem mengarahkan ke <i>user</i> untuk <i>login</i> aplikasi.
Postconditions	Halaman login ditampilkan

Tabel 30. *Use Case Proses Login.*

Use case ID	ST-002
Use case Name	<i>Login</i>
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	<i>User/Siswa</i>
Description	<i>Use case ini melakukan proses login untuk dapat menjalankan aplikasi dan mengakses soal yang disajikan.</i>
Pre-conditions	<i>Halaman Splash Screen</i>
Basic Path	<ol style="list-style-type: none"> 1. <i>Client/user</i> melakukan entry ID dan <i>password</i>. 2. Sistem akan mengarahkan ke halaman "Menu" pilih soal
Postconditions	Halaman login ditampilkan

Tabel 31. *Use Case Menu Pilih Soal.*

Use case ID	ST-003
Use case Name	Menu
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	<i>User/Siswa</i>
Description	<i>Use case ini melakukan proses menu pilih soal untuk dapat mengakses halaman soal untuk dikerjakan sesuai pilihan yang telah disajikan.</i>
Pre-conditions	<i>Halaman menu soal</i>
Basic Path	<ol style="list-style-type: none"> 1. <i>User</i> memilih soal mata pelajaran. 2. Sistem akan mengarahkan <i>user</i> ke daftar pilihan soal yang telah disajikan. 3. Sistem akan mengarahkan ke halaman soal.
Postconditions	Halaman Menu ditampilkan

Tabel 32. *Use Case Menu Input Jawaban.*

Use case ID	ST-004
Use case Name	Menu Test
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	<i>User/Siswa</i>
Description	<i>Use case ini menjelaskan proses mengerjakan soal dengan cara memilih jawaban benar sesuai pilihan jawaban telah tersedia pada halaman soal.</i>
Pre-conditions	<i>Halaman Input Jawaban</i>
Basic Path	<ol style="list-style-type: none"> 1. Sistem akan mengarahkan <i>user</i> ke halaman "Test" dan soal akan muncul sesuai mata pelajaran yang telah dipilih pada menu soal. 2. <i>User</i> memilih salah satu jawaban yang dianggap sesuai.
Postconditions	Halaman soal ditampilkan

Tabel 33. *Use Case* Proses *Upload* Nilai.

Use case ID	ST-005
Use case Name	<i>Upload</i> Nilai
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	User/Siswa
Description	<i>Use case</i> ini melakukan proses <i>upload</i> nilai (<i>scoring</i>) dengan cara <i>men-submit</i> jawaban yang telah dipilih dan dianggap benar pada halaman soal.
Pre-conditions	Halaman <i>Kerjakan Soal</i>
Basic Path	<ol style="list-style-type: none"> 1. Sistem akan meyakinkan <i>user</i> untuk <i>men-submit</i> jawaban yang dianggap benar melalui pesan. 2. Sistem akan mengarahkan <i>user</i> untuk <i>meng-upload</i> jawaban untuk dikonversi menjadi nilai (<i>scoring</i>).
Postconditions	Halaman <i>Upload</i>

Sistem Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* terintegrasi dengan aplikasi *web*. Berikut adalah penjelasan aktivitas yang dilakukan pada aplikasi *web*:

Tabel 34. *Use Case* *Login* Aplikasi Web

Use case ID	ST-006
Use case Name	<i>Login</i> Aplikasi <i>Web</i>
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	<i>Client/user</i> dan Guru/Admin
Description	<i>Use case</i> ini melakukan proses <i>login</i> aplikasi web untuk dapat melanjutkan menu dari aplikasi <i>mobile</i> . Menu aplikasi berfungsi menyajikan hasil sebagai evaluasi untuk <i>user</i> .
Pre-conditions	Halaman <i>front page</i>
Basic Path	<ol style="list-style-type: none"> 1. Halaman depan akan ditampilkan di <i>web browser</i>. 2. Sistem akan mengarahkan Siswa maupun Guru untuk <i>login</i> aplikasi web.
Postconditions	<i>Browser</i> menampilkan <i>front page</i> aplikasi web

Tabel 35. Use Case Melihat Nilai Aplikasi Web.

Use case ID	ST-007
Use case Name	Melihat Nilai
Use case Type	<i>Essential</i>
Priority	<i>High</i>
Actors	<i>Client/user</i> dan Guru/Admin
Description	Use case ini melakukan proses melihat nilai (<i>scoring</i>) sebagai hasil evaluasi <i>user</i> .
Pre-conditions	Halaman Tabel Nilai
Basic Path	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu Nilai 2. Sistem akan menampilkan nilai hasil mengerjakan soal pada aplikasi <i>mobile</i>. 3. Guru dapat melihat seluruh nilai siswa dalam bentuk tabel.
Postconditions	Browser menampilkan data nilai.

Tabel 36. Use Case Menu Pembahasan Aplikasi Web

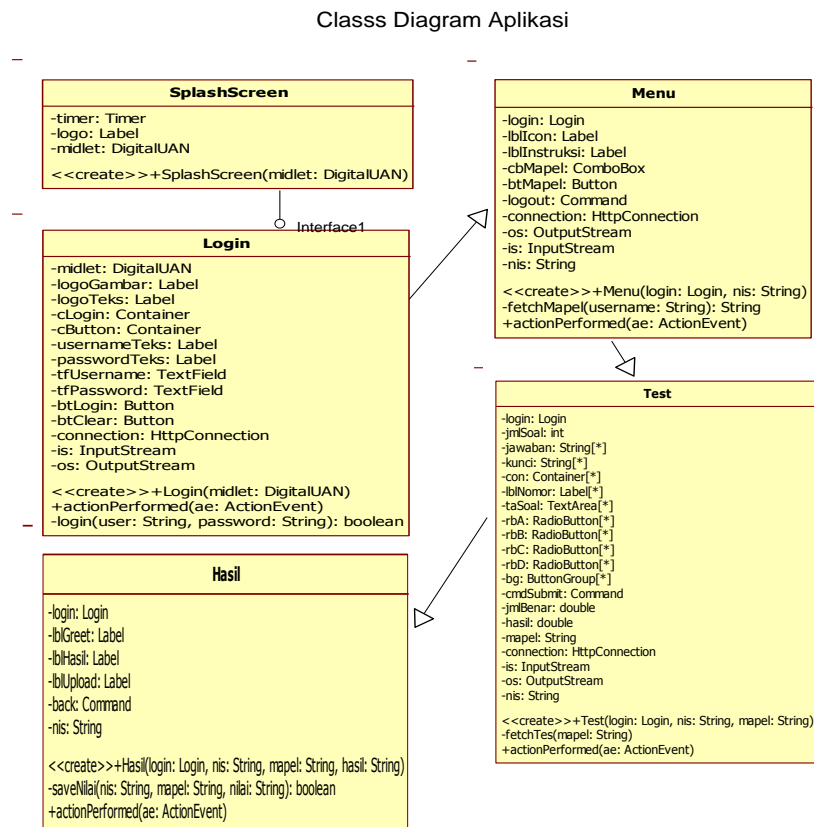
Use case ID	ST-008
Use case Name	Menu Pembahasan
Use case Type	<i>Not Essential</i>
Priority	<i>Low</i>
Actors	<i>User</i>
Description	Use case ini menampilkan pembahasan dari soal yang disajikan dalam aplikasi <i>mobile</i> .
Pre-conditions	Halaman Menu Pembahasan
Basic Path	<ol style="list-style-type: none"> 1. <i>Client/user</i> memilih menu pembahasan 2. Sistem akan menampilkan pembahasan dari soal yang telah dikerjakan <i>user</i> pada aplikasi <i>mobile</i>. 3. Guru dapat melihat hasil pembahasan dengan memilih fitur pembahasan.
Postconditions	Browser menampilkan jawaban dari soal.

Tabel 37. Use Case Menu Pengayaan Pada Aplikasi Web.

Use case ID	ST-009
Use case Name	Menu Pengayaan
Use case Type	<i>Not Essential</i>
Priority	<i>Low</i>
Actors	<i>User</i>
Description	Use case ini menampilkan menu pengayaan sebagai meningkatkan kemampuan belajar daripada <i>user</i> .
Pre-conditions	Halaman Menu Pengayaan
Basic Path	<ol style="list-style-type: none"> 1. <i>User</i> memilih menu pengayaan 2. Sistem akan menampilkan soal pengayaan untuk dikerjakan oleh <i>user</i>. 3. Sistem akan mengarahkan <i>user</i> untuk memilih jawaban yang dianggap tepat. 4. Guru dapat melihat menu pengayaan.
Postconditions	Browser menampilkan halaman soal pengayaan

b. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Rancangan *class diagram* pada aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* ini adalah:



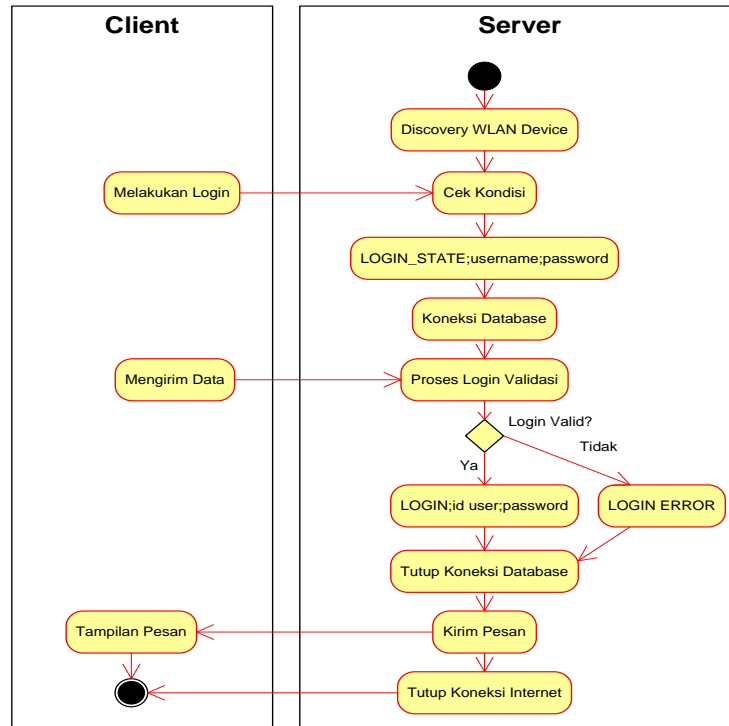
Gambar 12. Rancangan *Class Diagram*

c. Activity Diagram

Rancangan *Activity Diagram* aplikasi Simulasi Tes berbasis *mobile application* untuk *platform* Java adalah sebagai berikut:

1. Proses *login* aplikasi

Berikut ini adalah hasil rancangan *activity diagram* aplikasi Simulasi Tes berbasis *Mobile Application*:



Gambar 13. Activity Diagram Sequence Login

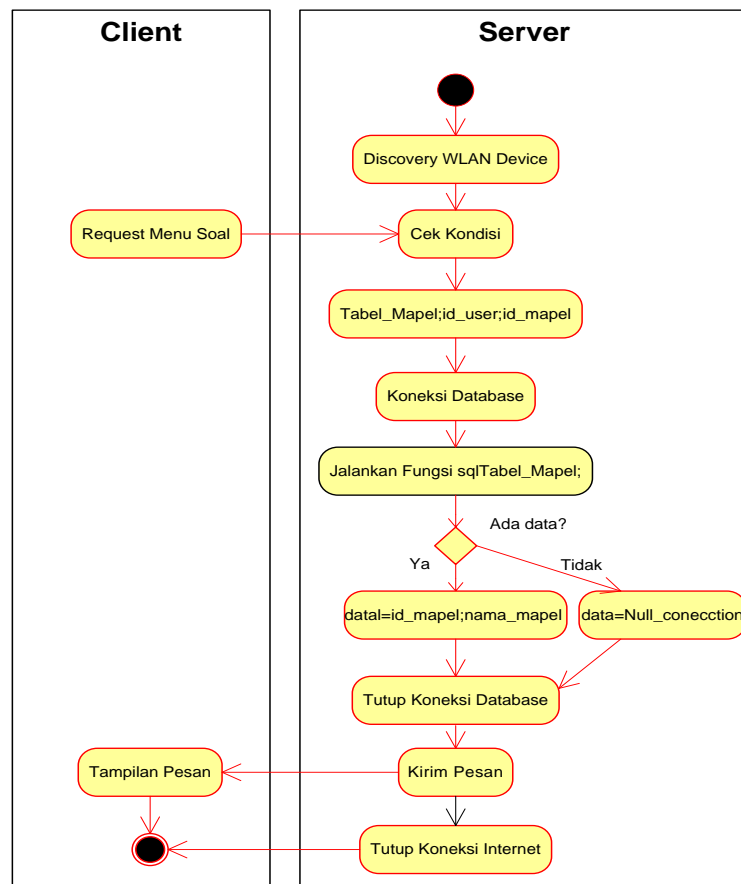
Pada saat koneksi *client-server* dibangun, proses yang dilakukan setelah koneksi berjalan adalah *user* (siswa) melakukan *login* yang bertujuan untuk identifikasi pada *user*. *User* akan mengirimkan data *login*, yaitu berupa *username* dan *password*. Data (*username* dan *password*) tersebut kemudian diolah oleh *server* dan divalidasi dengan *database*. Jika login valid, maka *server* akan mengirimkan data *id_user* dan *id_password*. Sedangkan jika proses *login* tidak sesuai, maka pesan yang dikirimkan *server* adalah *Login Error*.

2. Proses "Menu"

Pada proses "Menu", *client/user* memilih mata pelajaran yang diujikan pada aplikasi. Soal yang dipilih (*request*) *client/user* akan dikirim ke *server* dengan memanggil data pada *tabel_mapel*. *Request* yang dikirim menunjukkan bahwa *client/user* melakukan permintaan materi yang diujikan melalui fitur "Menu" dalam aplikasi Simulasi Tes.

Proses yang dilakukan yaitu, *client* akan mengirimkan *request message*, kemudian *server* menindaklanjuti dengan proses pemanggilan data soal yang dipilih dengan melakukan *query* ke *database*. Data dari *server* yang ada kemudian dikirimkan kepada *client* berupa *id_tipe*, *nama_tipe*.

Berikut ini adalah hasil rancangan *activity diagram* proses menu pilih soal aplikasi Simulasi Tes Berbasis *Mobile Application* yang ditunjukkan pada gambar 14:



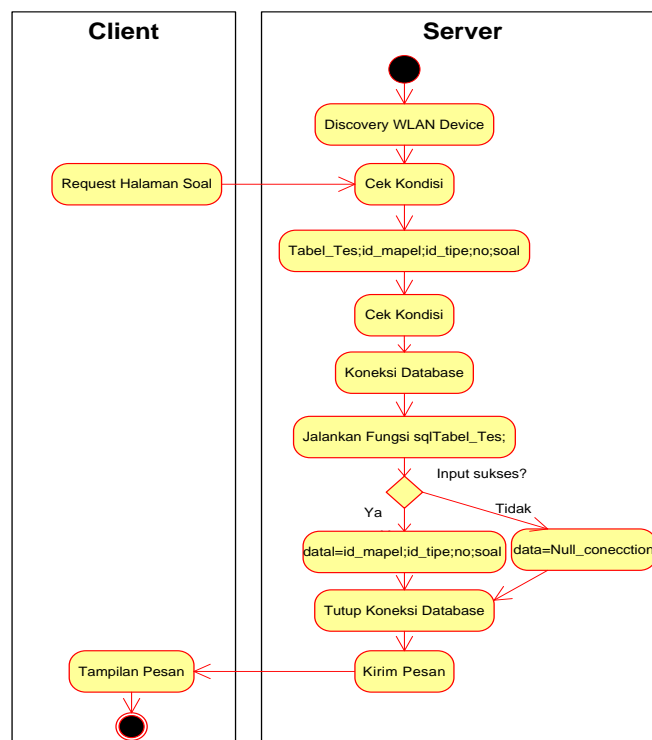
Gambar 14. *Activity Diagram* “Menu” Pilih Soal

3. Proses Menu “Test”

Pada saat proses mengerjakan soal simulasi tes pada aplikasi, terdapat beberapa data yang dikirimkan oleh *client*. Yaitu, data kondisi: *id_mapel*, *id_tipe*, *id_no*, *id_soal* serta pilihan jawaban *id_a*, *id_b*, *id_c*, *id_d* dan *id_e* untuk proses

input jawaban. Proses tersebut menunjukkan bahwa *client/user* melakukan permintaan jawaban melalui fitur “Test” dalam aplikasi Simulasi Tes.

Proses yang dilakukan yaitu, *client* akan mengirimkan *request message* pilihan jawaban yang telah dipilih untuk dikonversi menjadi sebuah nilai. Setelah itu, *server* menindaklanjuti *query* dari *client/user* dengan memvalidasi jawaban pilihan sesuai dengan jawaban yang benar, yaitu: id_jawaban ke *database*. Data dari *server* kemudian dikirim melalui *reply message* kepada *client* dan dikonversi menjadi data *string* nilai id_nilai. Apabila proses *input* jawaban terjadi kesalahan terhadap koneksi maupun paket data transfer, maka *server* akan mengirimkan “gagal upload”. Berikut ini adalah hasil rancangan *activity diagram* proses “Test”:



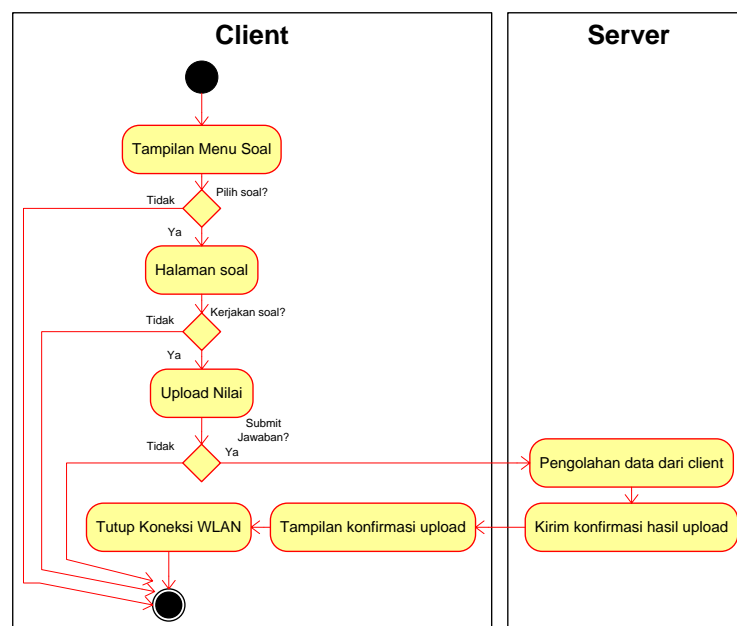
Gambar 15. Activity Diagram “Test”

4. Proses “Upload” Nilai

Proses “Upload” nilai *client/user* dilakukan setelah mengerjakan soal pada aplikasi. Data dikirimkan oleh *client/user* ke *server* berupa id_jawaban dan data

kondisi setelah proses *submit* jawaban, yaitu “*Upload*” nilai. Data (*request message*) yang dikirim ke *server* akan diterima dari *client* yang kemudian akan divalidasi dan akan dimasukan dalam *database*. *Server* akan mengolah data tersebut dan kemudian akan dikonversi menjadi data *integer* berupa *id_Nilai* dalam aplikasi web simulasi tes. Apabila proses *upload* nilai berjalan sukses, pada layar akan muncul pesan (*reply message*) dari *server*, yaitu “Nilai sudah diupload”. Sebaliknya, apabila terjadi kegagalan *upload* nilai, maka *server* akan memberikan *reply message*, yaitu “Nilai gagal diupload”.

Server akan mengirim *reply message* berupa nilai yang telah dikonversi oleh logika program dan akan ditampilkan dalam menu “Nilai” pada aplikasi web simulasi tes. Berikut adalah rancangan *activity diagram* Aplikasi Simulasi Tes berbasis *Mobile Application*:



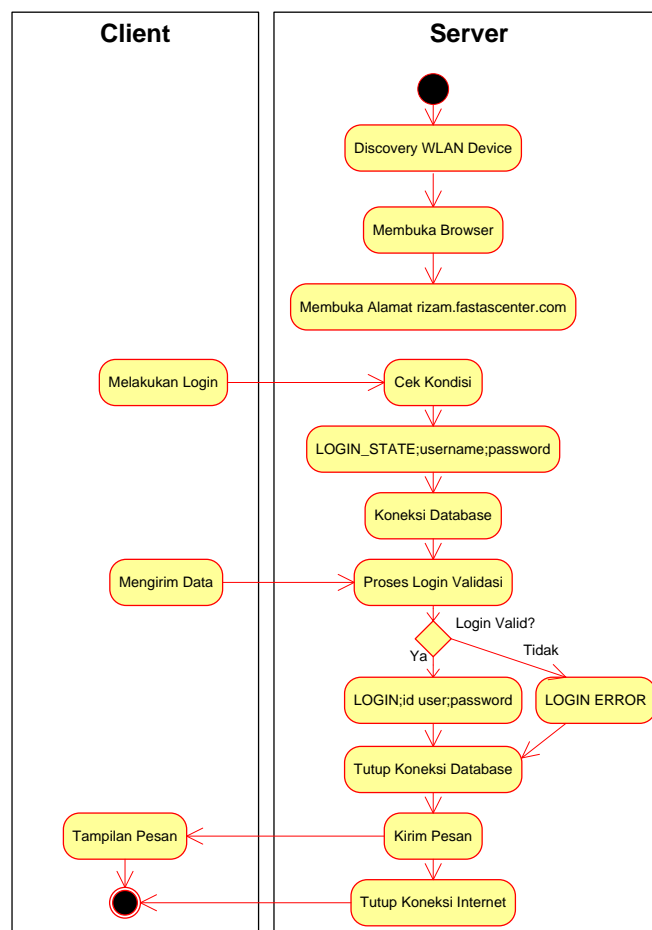
Gambar 16. Activity Diagram “Upload” Nilai

5. Proses “Login” Aplikasi Web

Pada proses *login* aplikasi web Simulasi Tes tidak jauh berbeda dengan Aplikasi *Mobile*. Proses yang dilakukan setelah *client/user* mengerjakan soal dan

logout dari aplikasi *mobile*, yaitu *login* pada aplikasi web Simulasi Tes. Pertama kali, *client/user* membuka *browser* dan membuka alamat <http://rizam.fastacenter.com>.

Proses yang dilakukan, yaitu *client/user* setelah koneksi berjalan adalah melakukan *login* yang bertujuan untuk verifikasi data pengguna. *Client/user* akan mengirimkan data *login* (*username* dan *password*) sama ketika akan *login* pada aplikasi *mobile* simulasi tes. Data tersebut kemudian diterima oleh *server* dan divalidasi dengan *database*. Jika *login* valid maka *server* akan mengirimkan data *id_user* dan *id_password*. Sedangkan, jika proses *login* tidak sesuai, maka pesan yang dikirimkan *server* adalah *Login Error*.

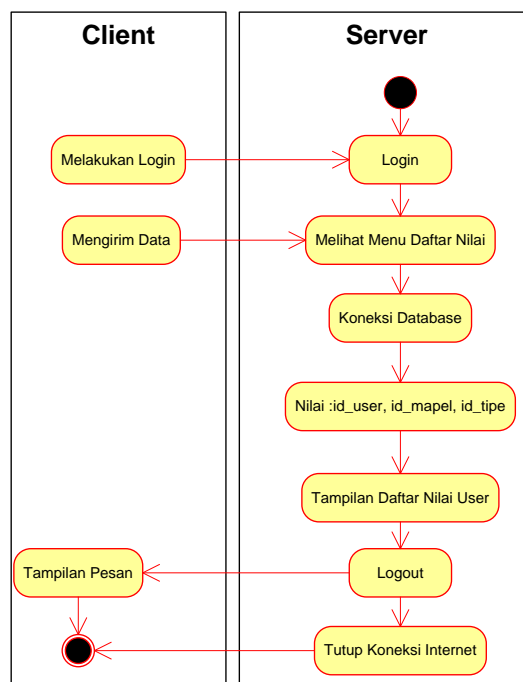


Gambar 17. Activity Diagram Login Aplikasi Web

6. Proses Menu “Nilai”

Pertama-tama, untuk melihat fitur “Nilai”, *client/user* menekan *button* menu “Nilai”. Data kondisi, *client/user* mengirimkan *request message* ke *server* yaitu permintaan untuk data id_Nilai siswa dari *database*. Hasil pemanggilan data nilai dari *database* kemudian dikirimkan ke *client/user*. Sehingga *client/user* akan dengan segera mendapatkan informasi mengenai nilai yang diperoleh setelah mengerjakan soal evaluasi.

Berikut ini adalah hasil rancangan *activity diagram* pada proses menu “Nilai” dapat dilihat pada gambar 18:

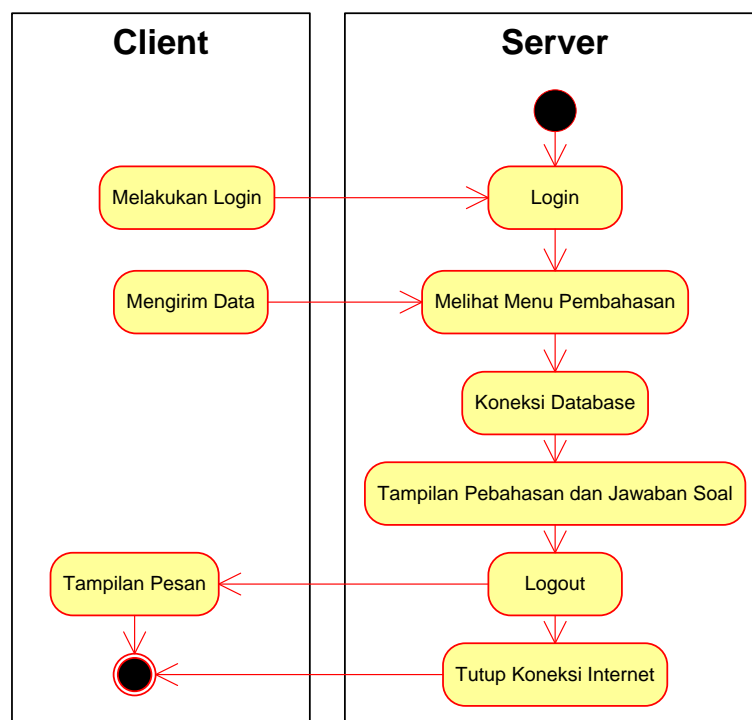


Gambar 18. Activity Diagram Menu “Nilai” Aplikasi Web

7. Proses Menu “Pembahasan”

Pada proses menu “Pembahasan”, beberapa data yang dikirimkan oleh *client/user*, yaitu data kondisi pembahasan. Pada saat *client/user* menekan *button* menu “Pembahasan”, maka kondisi *request message* yang dikirimkan ke *server* yaitu permintaan untuk data id_Mapel, id_tipe: no, pembahasan dari

database. Hasil pemanggilan data pembahasan dari *database* akan dikirimkan ke *client/user*. Sehingga *client/user* akan dengan mudah mendapatkan informasi mengenai jawaban yang benar setelah mengerjakan soal evaluasi. Berikut adalah hasil rancangan *activity diagram* menu “Pembahasan” yang ditunjukkan pada gambar 19:

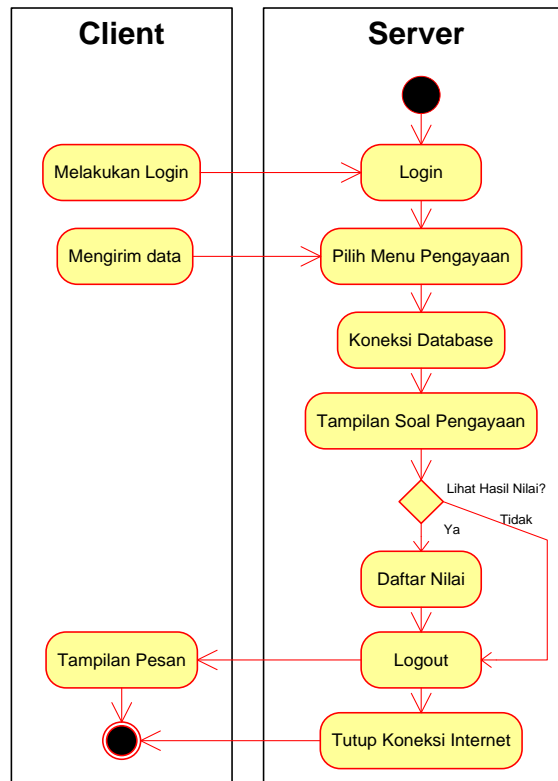


Gambar 19. *Activity Diagram* Menu Pembahasan

8. Menu “Pengayaan”

Proses menu “Pengayaan” terdapat beberapa data yang dikirimkan oleh *client/user*, yaitu data kondisi pengayaan. Pada saat *client/user* menekan *button* menu “Pembahasan”, maka kondisi *request message* yang dikirimkan ke *server*, yaitu permintaan untuk data *id_tipe*, *no_tipe* dari *database*. Hasil pemanggilan data pengayaan dari *database* akan dikirimkan ke *client/user*. Sehingga *client/user* dapat melakukan pengayaan untuk menambah penilaian dalam

mengikuti simulasi tes. Berikut adalah hasil rancangan *activity diagram* menu “Pengayaan” yang ditunjukkan pada gambar 20:



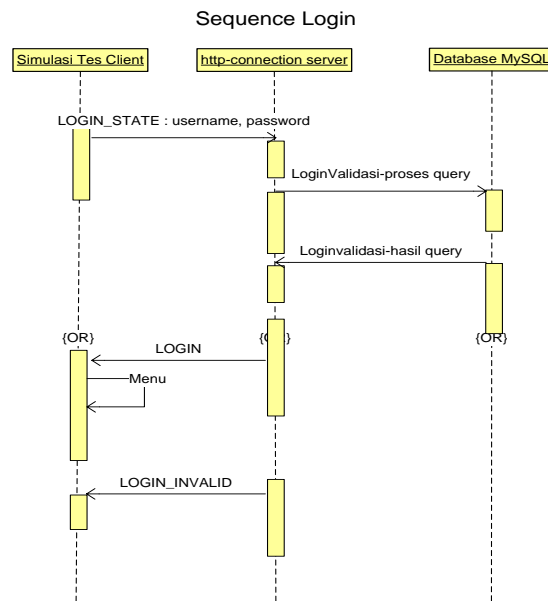
Gambar 20. *Activity Diagram* Menu Pengayaan

d. *Sequence Diagram*

Rancangan awal *sequence diagram* aplikasi Simulasi Tes berbasis *mobile application* adalah sebagai berikut:

1. *Sequence* “Login”

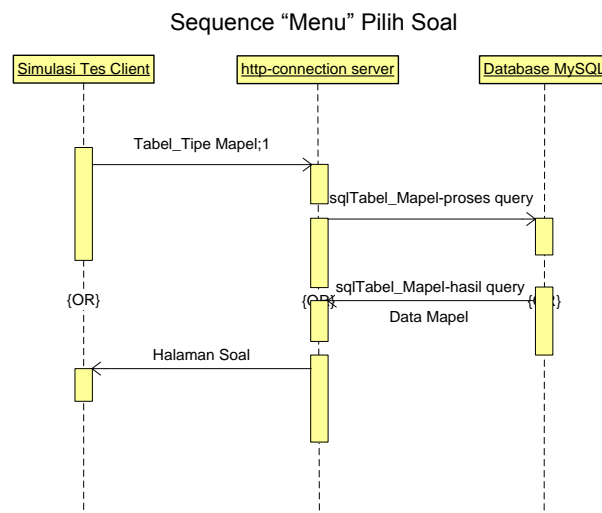
Dari gambar 21 dapat dilihat bahwa pertama-tama *client/user* akan mengirimkan *username* dan *password* kepada *server* dalam bentuk *string*. Selanjutnya, pesan yang diterima *server* akan diproses ke *database*. Pada proses tersebut akan dilakukan verifikasi data oleh server sesuai *database* yang benar. Berikut adalah hasil rancangan *sequence* “login”:



Gambar 21. Sequence "Login"

2. Sequence "Menu" Soal

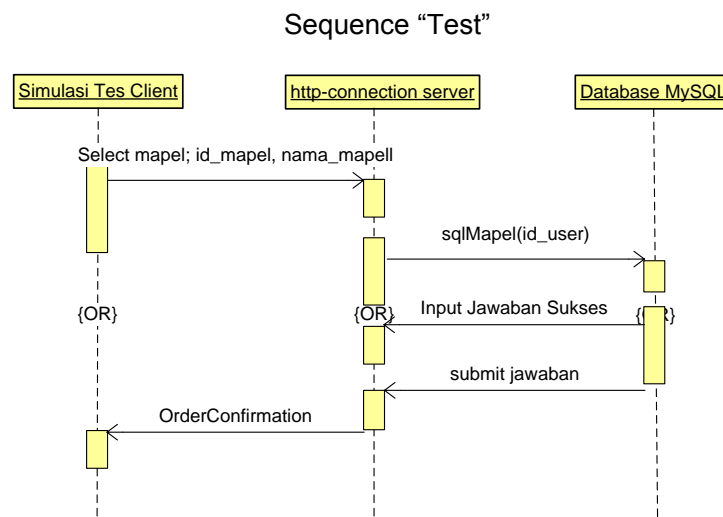
Pada gambar 22 di bawah menunjukkan *message sequence* dari proses permintaan menu soal yang aktif pada *http-connection server* dari *client/user* (*request message*). *Http-connection server* akan mengirimkan *reply message* berupa data soal. Apabila pada saat *client/user* mengirimkan *request message* gagal, maka *client/user* akan menampilkan pesan *error*.



Gambar 22. Sequence "Menu" Pilihan Soal

3. Sequence “Test”

Pada proses menu “Test”, *client/user* mengirimkan data input jawaban. Data jawaban diterima oleh *http-connection server*, kemudian *server* mengolah data jawaban dan dikonversi menjadi data nilai siswa. Sebagai pesan balasan (*reply message*) dari *http-connection server*, yaitu data informasi berupa *id_user*: *mapel*, *tipe*, *nilai* yang kemudian dikonversi menjadi nilai oleh logika program dan ditampilkan pada menu “Nilai” aplikasi web simulasi tes. Berikut adalah *Message sequence* “Test” ditunjukkan pada gambar 23.

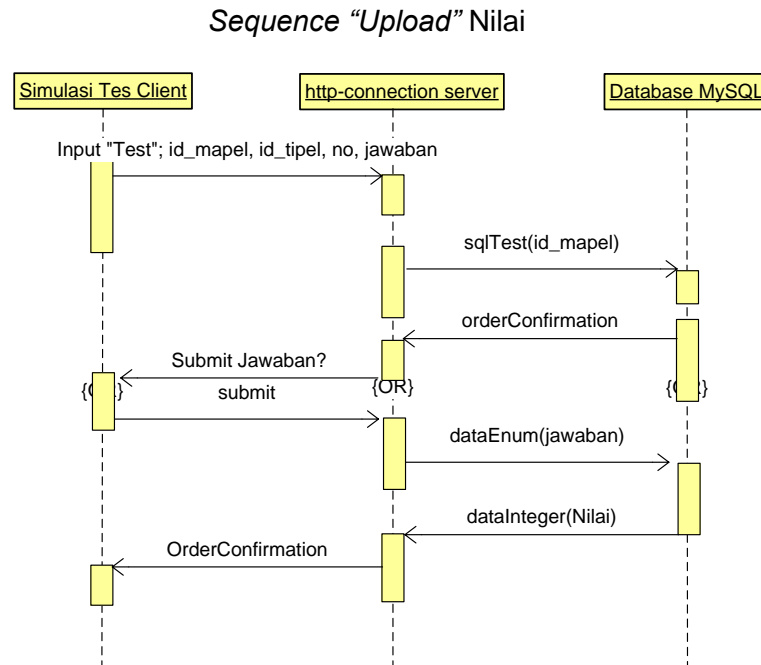


Gambar 23. Sequence Menu “Test”

4. Sequence “Upload” Nilai

“Upload” nilai merupakan proses terakhir setelah *client/user* mengerjakan soal pada aplikasi *mobile*. “Upload” nilai adalah tahap di mana *client/user* melakukan proses *submit* jawaban yang telah diinput. Sebelum melakukan *submit* jawaban, *client/user* melakukan *request* input jawaban pada menu “Test”. Data input tersebut kemudian dikirimkan ke server berupa, yaitu *id_jawaban*. Program akan mengirimkan *reply message* berupa konfirmasi proses *submit* jawaban yaitu “Upload” nilai. Server akan menerima *request message* dari *client* yang kemudian akan divalidasi dan akan dimasukkan dalam *database*. Server

akan mengolah data tersebut dan kemudian akan dikonversi menjadi data *integer* berupa *id_Nilai* dalam aplikasi web Simulasi Tes. Berikut adalah rancangan sequence “*Upload*” *nilai* yang ditunjukkan pada gambar 24:

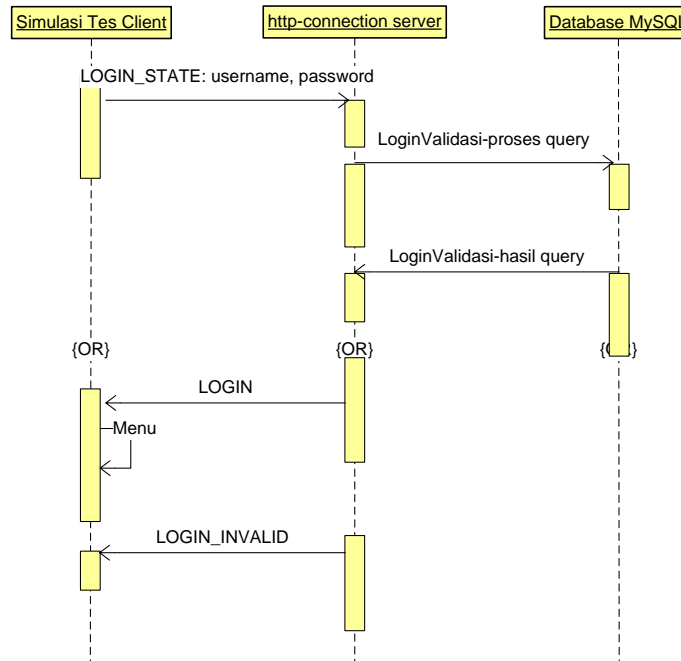


Gambar 24. Sequence “*Upload*” Nilai

5. Sequence “*Login*” Aplikasi Web

Proses *login* pada Aplikasi Web simulasi tes dilakukan pasca mengerjakan soal yang terdapat pada aplikasi *mobile* simulasi tes. Dari gambar 24 dapat terlihat bahwa untuk masuk dalam aplikasi kedua yang berbasis web (*web base*), pertama-tama *client/user* akan mengisi data inputan berupa *username* dan *password* yang sama saat login pada aplikasi *mobile*.

Sequence Login Aplikasi Web

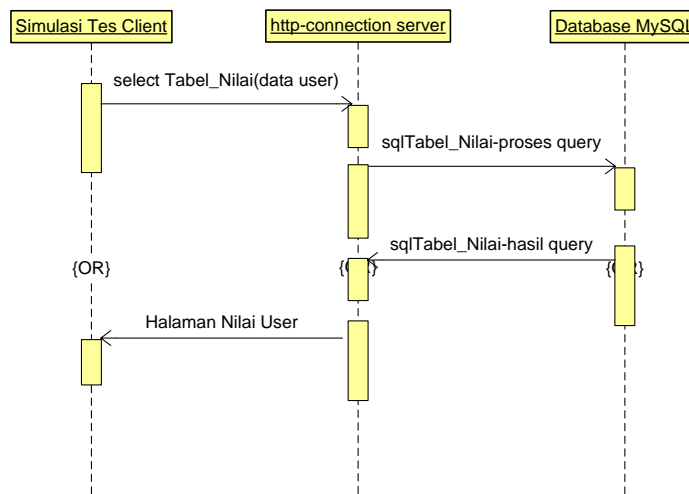


Gambar 25. Sequence “Login” Aplikasi Web

6. Sequence Menu “Nilai”

Pada gambar 26 menunjukan hasil pengolahan skor jawaban setelah *clien/user* mengerjakan soal pada aplikasi. Jawaban dari *clien/user* akan dikirimkan ke *server* dengan ID Tabel_Nilai. Hasil jawaban yang dikerjakan melalui permintaan/*message* dari *clien/user* akan dikirimkan ke *database* oleh *http-connection server*. *Http-connection server* akan menjalankan fungsi `sqlSelectNilai()` untuk mengirimkan *message* berupa hasil jawaban yang telah dikonversi menjadi nilai oleh logika program ke *database*. *Database* mengirimkan *reply message* berupa data nilai sesuai permintaan *user*. Sequence diagram menu “Nilai” yang ditunjukan pada gambar berikut:

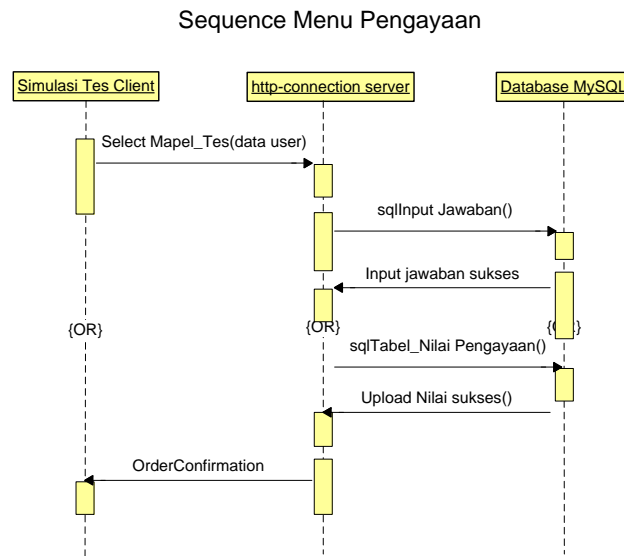
Sequence Menu Nilai



Gambar 26. Sequence Menu “Nilai”

7. Sequence Menu “Pengayaan”

Perancangan aplikasi Simulasi Tes dengan mengadopsi penggunaan nilai pengayaan. Fitur menu “Pengayaan” berfungsi sebagai evaluasi untuk mengarah penilaian yang lebih obyektif. Menu pengayaan secara mekanisme program akan muncul apabila *client/user* dalam mengerjakan soal mendapat nilai kurang di bawah standar. Pada proses ini *http-connection server* akan memproses *message* ke *database* dan memanggil data *id_mapel (tipe_2)*. Untuk *message sequence*-nya dapat dilihat dari pada gambar 27 sebagai berikut:

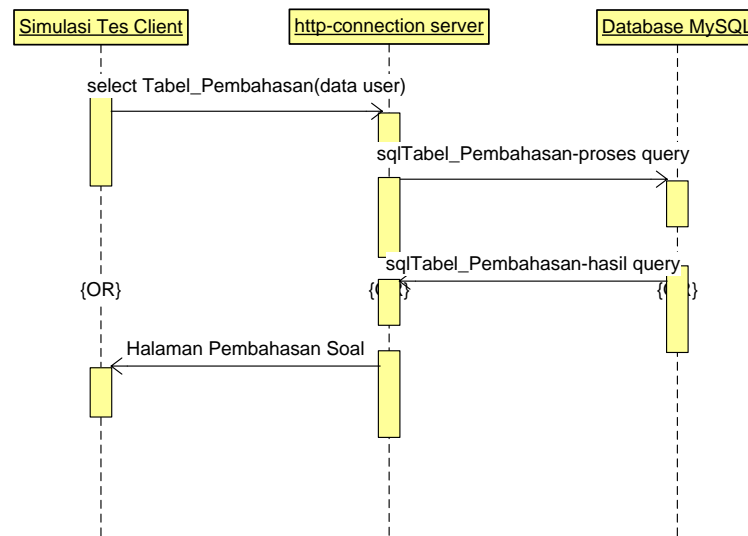


Gambar 27. Sequence Menu Pengayaan

8. Sequence Menu “Pembahasan”

Setelah melakukan pengerjaan soal pada aplikasi *mobile*, *client/user* dapat melakukan fungsi melihat hasil nilai dan melihat hasil evaluasi berupa jawaban yang benar. Proses menampilkan pembahasan pada MIDlet *client*, pertama-tama *client/user* akan memilih menu “Pembahasan” pada aplikasi web melalui *browser*. Permintaan dari *client/user* kemudian dikirimkan oleh MIDlet *client* ke *http-connection server* untuk mendapatkan jawaban yang benar dari menu “Pembahasan”. *Message sequence* dari proses ini ditunjukkan pada gambar 28. *Http-connection server* akan melakukan proses *query* untuk mengambil data menu pembahasan pada *database*. Jawaban yang benar akan dikirimkan pada MIDlet *client* sebagai *reply message*.

Sequence Menu Pembahasan



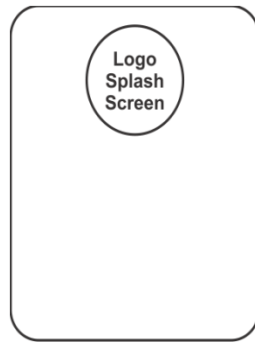
Gambar 28. Sequence Menu Pembahasan

e. Perancangan *User Interface*

Rancangan awal *user interface* aplikasi Simulasi Tes berbasis *mobile application* adalah sebagai berikut:

1. Rancangan Tampilan Awal *Splash Screen*

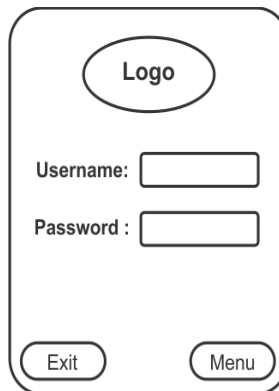
Tampilan awal *splash screen* adalah hasil eksplorasi tampilan grafis di awal pembuka sebuah aplikasi. *Style splash screen* banyak digunakan untuk memberikan identitas sebuah *software* (perangkat lunak). Berikut adalah rancangan tampilan *splash screen* untuk aplikasi berbasis *mobile application* yang ditunjukkan pada gambar 29.



Gambar 29. Rancangan Tampilan Awal *Splash Screen*

2. Rancangan *Form Login*

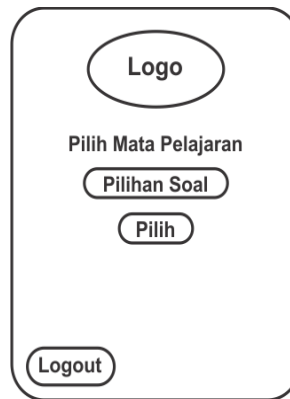
Tampilan kedua setelah adalah halaman *form login* berisi 2 buah *textfield* yang berisikan *username* dan *password*. Dengan menyesuaikan resolusi layar yang berbeda-beda, akan memudahkan *user* melakukan navigasi pada *handphone*. Berikut adalah rancangan tampilan *form login* untuk aplikasi Simulasi Tes berbasis *mobile application* yang ditunjukkan pada gambar 30.



Gambar 30. Rancangan Tampilan Halaman *Login*

3. Rancangan Tampilan Menu Pilih Soal

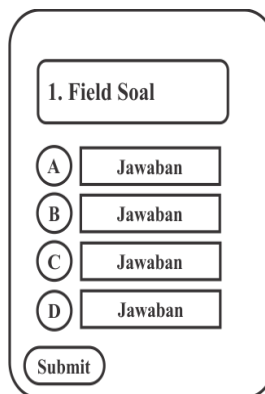
Tampilan menu pilih soal akan menampilkan daftar soal mata pelajaran yang akan dikerjakan oleh *user*. *User* terlebih dahulu memilih mata pelajaran sebelum memulai mengerjakan soal. Berikut adalah rancangan tampilan menu pilih soal untuk aplikasi Simulasi Tes berbasis *Mobile Application* yang ditunjukkan pada gambar 31.



Gambar 31. Rancangan Tampilan Halaman Pilih Soal

4. Rancangan Halaman Soal

Tampilan halaman kerja soal menampilkan soal beserta pilihan jawaban menggunakan *textfield* yang dimodifikasi dengan perintah *running text* untuk menyesuaikan resolusi layar yang berbeda serta memudahkan *client/user* dalam mengamati soal. Berikut adalah rancangan tampilan halaman soal untuk aplikasi Simulasi Tes berbasis *mobile application* yang ditunjukkan pada gambar 32.

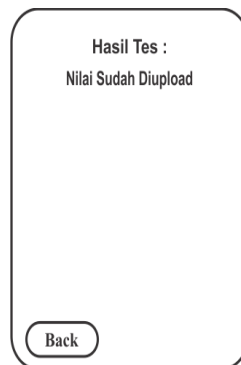


Gambar 32. Rancangan Tampilan Halaman Soal

5. Rancangan Tampilan *Upload* Nilai

Rancangan tampilan *upload* nilai akan muncul apabila *user* sudah merasa yakin dengan meng-*input* jawaban yang dianggap benar. Bentuk pesan dengan menggunakan *alert message* akan meyakinkan *user* untuk proses *submit* jawaban yang akan dikonversi menjadi nilai (*scoring*). Berikut adalah rancangan

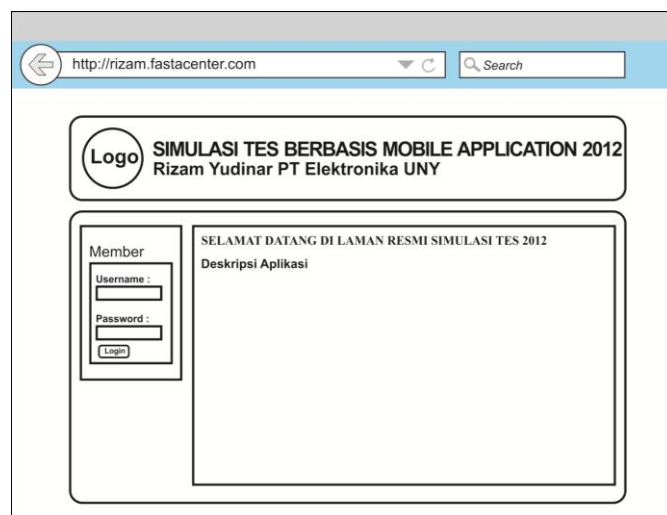
tampilan *upload nilai* untuk aplikasi Simulasi Tes Berbasis *Mobile Application* yang ditunjukkan pada gambar 33.



Gambar 33. Rancangan Tampilan *Upload Nilai*

6. Rancangan Tampilan “*Login*” Aplikasi *Web*

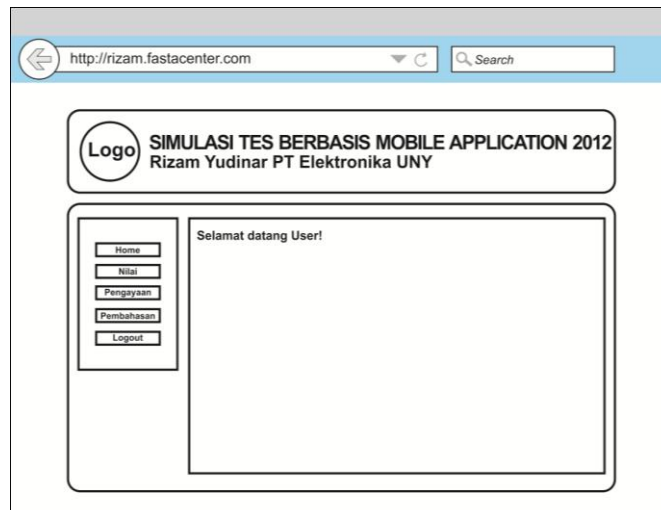
Rancangan tampilan *login* pertama akan muncul pada *browser* apabila *client/user* mengunjungi alamat <http://rizam.fastacenter.com>. *Client/user* dapat melanjutkan proses *login* agar dapat menjalankan fitur yang telah disediakan dalam sistem. Berikut adalah rancangan tampilan *login* untuk aplikasi Simulasi Tes berbasis *mobile application* yang ditunjukkan pada gambar 34.



Gambar 34. Rancangan Tampilan “*Login*”

7. Rancangan Tampilan Menu Utama

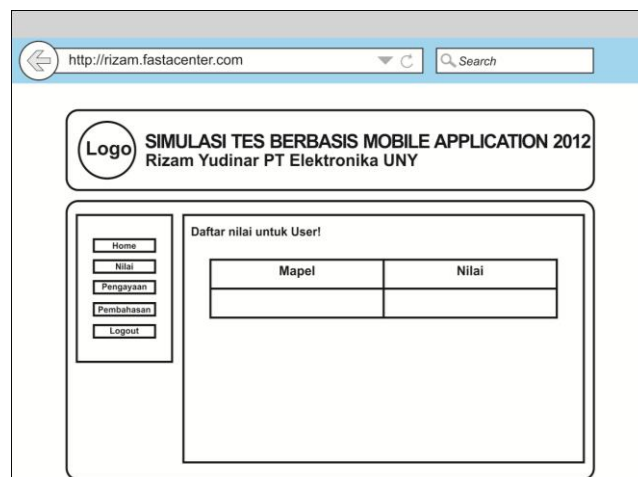
Rancangan tampilan menu utama akan muncul setelah *client/user* melakukan *login* pada aplikasi *web* simulasi tes. Berikut adalah rancangan tampilan menu utama yang ditunjukkan pada gambar 35.



Gambar 35. Rancangan Tampilan Menu Utama

8. Rancangan Tampilan Menu Nilai

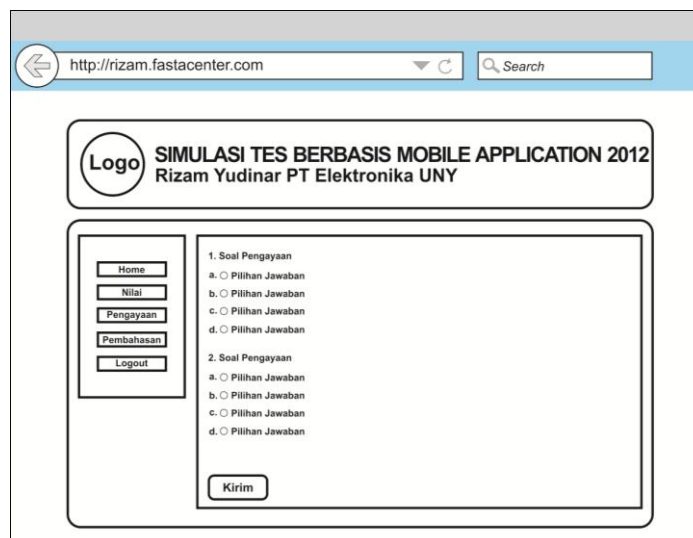
Rancangan menu “Nilai” difungsikan untuk melihat nilai (*scoring*) setelah mengerjakan soal utama pada aplikasi *mobile* maupun mengerjakan soal pengayaan yang terdapat pada aplikasi *web* simulasi tes. Berikut adalah rancangan tampilan menu nilai yang ditunjukkan pada gambar 36.



Gambar 36. Rancangan Tampilan Menu Nilai

9. Rancangan Tampilan Menu Pengayaan

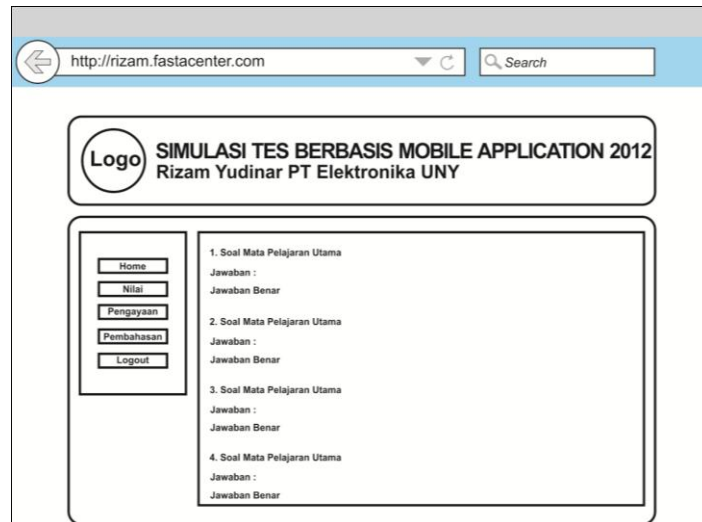
Rancangan tampilan menu pengayaan difungsikan apabila nilai *client/user* menurut guru masih dirasa kurang memenuhi standar. *Client/user* akan mengakses fitur tersebut untuk perbaikan menuju standar yang ditetapkan guru. Berikut adalah rancangan tampilan menu pengayaan yang ditunjukkan pada gambar 37.



Gambar 37. Rancangan Tampilan Menu “Pengayaan”

10. Rancangan Tampilan Menu Pembahasan

Rancangan tampilan menu “Pembahasan” difungsikan dan muncul pada aplikasi *Web Simulasi Tes* setelah *Client/user* mengerjakan soal pada aplikasi *mobile*. Menu “Pembahasan” menyajikan pembahasan jawaban yang tepat dan benar sesuai soal yang disajikan. Berikut ini adalah rancangan tampilan menu “Pembahasan” yang ditunjukkan pada gambar 38.



Gambar 38. Rancangan Tampilan Menu “Pembahasan”

3. Desain Database

Proses implementasi *database* dalam mengembangkan aplikasi menggunakan *MySQL*. Berikut adalah hasil implementasi struktur *database* aplikasi Simulasi Tes Berbasis *Mobile Application* ditunjukkan pada tabel 38 sebagai berikut:

Tabel 38. Struktur Tabel Pada Database

Table	Action	Rows	Type	Collation	Size	Overhead
mapel	Browse Structure Search Insert Empty Drop	3	MyISAM	latin1_swedish_ci	2.1 KiB	-
nilai	Browse Structure Search Insert Empty Drop	32	MyISAM	latin1_swedish_ci	3.5 KiB	42 B
pembahasan	Browse Structure Search Insert Empty Drop	40	MyISAM	latin1_swedish_ci	7.1 KiB	-
tes	Browse Structure Search Insert Empty Drop	143	MyISAM	latin1_swedish_ci	40.2 KiB	1.4 KiB
tipe	Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci	2.0 KiB	-
user	Browse Structure Search Insert Empty Drop	80	MyISAM	latin1_swedish_ci	7.7 KiB	-
6 tables	Sum	300	InnoDB	latin1_swedish_ci	62.6 KiB	1.5 KiB

Berdasarkan implementasi, proses pengembangan merancang struktur *database* dengan 6 tabel. Setiap tabel memiliki peran dan fungsi masing-masing. Berikut penjelasan dari masing-masing tabel:

a. Tabel “Mapel”

Tabel 39. Detail Tabel Mapel

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	<u>id_mapel</u>	char(3)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	nama_mapel	varchar(30)	latin1_swedish_ci		No	None	

b. Tabel “Nilai”

Tabel 40. Detail Tabel Nilai

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id_user	char(5)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	id_mapel	char(3)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 3	id_tipe	enum('1', '2')	latin1_swedish_ci		No	1	
<input type="checkbox"/> 4	nilai	int(11)			No	None	

c. Tabel “Pembahasan”

Tabel 41. Detail Tabel Pembahasan

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id_mapel	char(3)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	id_tipe	enum('1', '2')	latin1_swedish_ci		No	1	
<input type="checkbox"/> 3	no	int(11)			No	None	
<input checked="" type="checkbox"/> 4	pembahasan	text	latin1_swedish_ci		No	None	

d. Tabel “Tes”

Tabel 42. Detail Tabel Tes

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id_mapel	char(3)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	id_tipe	enum('1', '2')	latin1_swedish_ci		No	None	
<input type="checkbox"/> 3	no	tinyint(4)			No	None	
<input type="checkbox"/> 4	soal	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 5	a	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 6	b	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 7	c	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 8	d	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 9	e	text	latin1_swedish_ci		No	None	
<input type="checkbox"/> 10	jawaban	enum('a', 'b', 'c', 'd', 'e')	latin1_swedish_ci		No	None	

e. Tabel “Tipe”

Tabel 43. Detail Tabel Tipe

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id_tipe	enum('1', '2')	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	nama_tipe	enum('pengayaan', 'utama')	latin1_swedish_ci		No	None	

f. Tabel “User”

Tabel 44. Detail Tabel User

#	Column	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	id_user	char(5)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 2	password	varchar(32)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 3	nama	varchar(50)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 4	alamat	varchar(50)	latin1_swedish_ci		No	None	
<input type="checkbox"/> 5	tipe	enum('siswa', 'guru')	latin1_swedish_ci		No	None	

C. *Development/Pengembangan*

Proses pengembangan aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* dilakukan IDE (*Integrated Development Environment*) Netbean 7.1 dengan menggunakan SDK (*Software Development Kit*) dari oracle mulai dari versi MIDP 1.0, 2.0 dan 2.1 ter-*install* dengan baik. Proses pengembangannya dilakukan sebagai berikut:

- a. Implementasi layout aplikasi menggunakan *script XML* dan didukung *library LWUIT*.
- b. Implementasi isi materi soal menggunakan bahasa pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.
- c. Implementasi logika program menggunakan bahasa pemrograman *Java*.
- d. Implementasi *user interface*.

Semua layout yang bukan gambar dikembangkan dengan menggunakan *script XML* sesuai syntax standar yang dikembangkan oleh Google. Layout yang dikembangkan sebagai kebutuhan aplikasi Simulasi Tes untuk *platform Java* di antaranya *project.xml* dan *build-impl.xml*. File-file tersebut belum termasuk *script XML* seperti *private.xml* serta berbagai file yang ada dalam *folder build*:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://www.netbeans.org/ns/project/1">
  <type>org.netbeans.modules.kjava.j2meproject</type>
  <configuration>
    <data xmlns="http://www.netbeans.org/ns/j2me-project">
      <name>digitalUAN</name>
      <minimum-ant-version>1.6</minimum-ant-version>
    </data>
  </configuration>
</project>
```

Gambar 39. Potongan *Script XML*

1. Implementasi Isi Soal Materi Menggunakan PHP dan MySQL

Pengembangan materi pembelajaran aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* menggunakan *script* Java dan PHP. PHP dipilih sebagai integrasi *database* yang diletakan pada *server* yang berbeda. Materi diolah menggunakan perintah *MySQL* menjadikan aplikasi bersifat dinamis. Berikut adalah potongan *source code* dari implementasi program menggunakan bahasa pemrograman PHP:

```
<?php
include_once 'initial.php';
$username = $_GET['username'];
$eng = mysql_query("SELECT * FROM nilai WHERE
id_user='".$username.'" AND id_mapel='eng' AND id_tipe='2'");
if(mysql_num_rows($eng)>0){
    echo "|no|";
}else{
    echo "|eng|";
}
$ind = mysql_query("SELECT * FROM nilai WHERE
id_user='".$username.'" AND id_mapel='fis' AND id_tipe='2'");
if(mysql_num_rows($ind)>0){
    echo "|no|";
}else{
    echo "|fis|";
}
$mat = mysql_query("SELECT * FROM nilai WHERE
id_user='".$username.'" AND id_mapel='kim' AND id_tipe='2'");
if(mysql_num_rows($mat)>0){
    echo "|no|";
}else{
    echo "|kim|";
}
?>
```

Gambar 40. Potongan Script PHP

Selain kelebihan di atas, *MySQL* juga memiliki kelebihan ke dalam aspek pengelolaan *user* sehingga keamanan aplikasi tetap terjaga.

2. Implementasi Logika Program Menggunakan Bahasa Pemrograman Java.

Layout dan materi yang telah dikembangkan tidak akan berjalan sebagai sebuah aplikasi tanpa adanya peran serta Java sebagai bahasa pemrograman utama pengembangan yang digunakan di *Java Micro Edition*. Java berperan dalam menangani aktivitas yang ada pada *layout* maupun dari sistem internal secara umum. Contoh penggunaan Java ditunjukkan pada pemberian *command button* yang ada pada *layout XML* agar dapat ditekan (diklik) dan kemudian melakukan navigasi ke halaman berikutnya.

3. Implementasi User Interface

Implementasi *user interface* aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* tidak terlepas dengan proses desain antarmuka program . Berikut ini merupakan hasil implementasi desain aplikasi Simulasi Tes untuk *platform* Java pada emulator.

a. Implementasi Desain Halaman *Splash Screen*

Proses untuk menjalankan aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* adalah diawali dengan munculnya logo atau sering disebut proses *splash screen*. Berikut adalah script untuk mengembangkan *spalsh screen*:

```

1 //tampilan Splash Screen//
2 import com.sun.lwuit.Component;
3 import com.sun.lwuit.Form;
4 import com.sun.lwuit.Image;
5 import com.sun.lwuit.Label;
6 import com.sun.lwuit.layouts.BorderLayout;
7 import java.io.IOException;
8 import java.util.Timer;
9 import java.util.TimerTask;
10
11 /*
12  * To change this template, choose Tools | Templates
13  * and open the template in the editor.
14  */
15
16 /**
17  *
18  * @author Ade
19  */
20 public class SplashScreen extends Form{
21     private Timer timer;
22     private Label logo;
23     private DigitalUAN midlet;
24
25     public SplashScreen(final DigitalUAN midlet){
26         this.midlet = midlet;
27         logo = new Label();
28         try {
29             logo.setIcon(Image.createImage("/img/logo.png"));
30         } catch (IOException ex) {
31             ex.printStackTrace();
32         }
33     }

```

Gambar 41. Potongan *Script* Halaman *Splash Screen*

Penggalan *source code* yang ditunjukkan pada gambar 41 akan menampilkan halaman *splash screen* untuk mengawali penggunaan aplikasi Simulasi Tes *mobile*. Berikut adalah tampilan proses *splash screen* menggunakan *emulator* dari *Sun microsystem*:

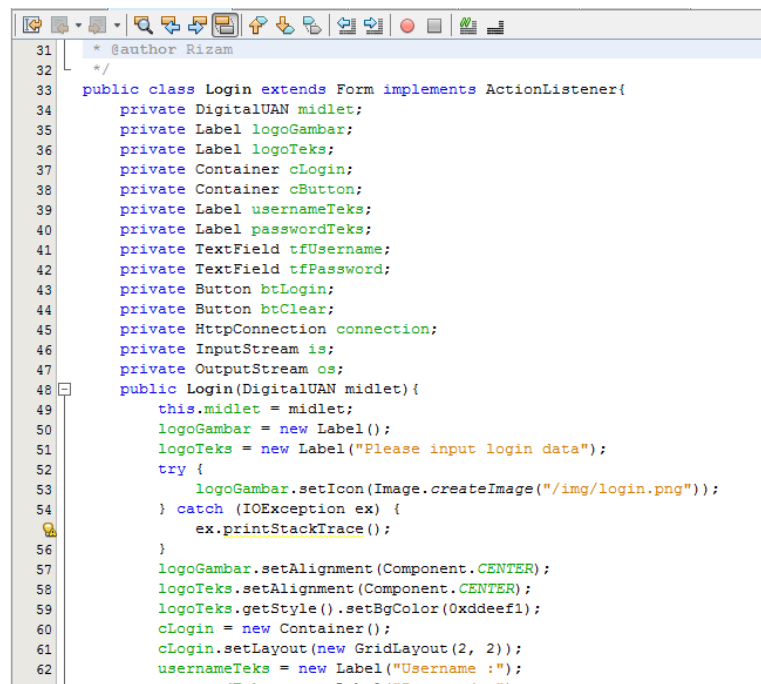


Gambar 42. Implementasi Halaman *Splash Screen*

b. Implementasi Halaman *Login*

Setelah *client/user* melewati proses halaman *splash screen*, kemudian dihadapkan dengan tampilan *form login*. Proses ini diawali oleh *client/user* pengisian *form login* atau mengirimkan *id_user*. Yaitu, *username* dan *password* kepada *server* dan melalui proses validasi agar dapat masuk dalam aplikasi Simulasi Tes *mobile*.

Berikut adalah penggalan *source code* untuk melakukan proses *login* pada aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*:



```
31  * @author Rizam
32  */
33  public class Login extends Form implements ActionListener{
34      private DigitalUAN midlet;
35      private Label logoGambar;
36      private Label logoTeks;
37      private Container cLogin;
38      private Container cButton;
39      private Label usernameTeks;
40      private Label passwordTeks;
41      private TextField tfUsername;
42      private TextField tfPassword;
43      private Button btLogin;
44      private Button btClear;
45      private HttpURLConnection connection;
46      private InputStream is;
47      private OutputStream os;
48      public Login(DigitalUAN midlet){
49          this.midlet = midlet;
50          logoGambar = new Label();
51          logoTeks = new Label("Please input login data");
52          try {
53              logoGambar.setIcon(Image.createImage("/img/login.png"));
54          } catch (IOException ex) {
55              ex.printStackTrace();
56          }
57          logoGambar.setAlignment(Component.CENTER);
58          logoTeks.setAlignment(Component.CENTER);
59          logoTeks.getStyle().setBgColor(0xdddeef1);
60          cLogin = new Container();
61          cLogin.setLayout(new GridLayout(2, 2));
62          usernameTeks = new Label("Username :");
63          passwordTeks = new Label("Password :");
```

Gambar 43. Potongan Script Halaman *Login*

Berikut hasil implementasi dari penggalan *source code* yang ditunjukkan pada gambar 43 akan menampilkan halaman *login* untuk masuk ke aplikasi Simulasi Tes *mobile* menggunakan *emulator* dari *Sun microsystem*:



Gambar 44. Implementasi Halaman Form Login

c. Implementasi Halman Tampilan Menu Soal

Tampilan halaman menu soal akan muncul setelah *client/user* melewati proses *login*. Aktivitas *client/user* pada halaman ini adalah memilih opsi pilihan soal yang disajikan dalam sistem. Pilihan soal dimunculkan berdasarkan data *id_mapel* dan *nama_mapel* dengan melakukan *request message* kepada *server* aplikasi Simulasi Tes *mobile*.

Hasil tampilan menunjukkan bahwa *client/user* memberikan *command* (perintah) melalui *button/tombol* “Pilih”. Pada perintah ini *client/user* terlebih dahulu memilih soal mata pelajaran menggunakan tombol *joypad* pada *handphone*. Berikut penggalan *source code* untuk memunculkan tampilan halaman “Menu” soal:

```

20  * @author Rizam
21  */
22  public class Menu extends Form implements ActionListener{
23      private Login login;
24      private Label lblIcon;
25      private Label lblInstruksi;
26      private ComboBox cbMapel;
27      private Button btMapel;
28      private Command logout;
29      private HttpConnection connection;
30      private OutputStream os;
31      private InputStream is;
32      private String nis;
33  public Menu(Login login, String nis){
34      this.login = login;
35      this.nis = nis;
36      setTitle("Menu");
37      setLayout(new BorderLayout(BoxLayout.Y_AXIS));
38      lblIcon = new Label();
39      try {
40          lblIcon.setIcon(Image.createImage("/img/mapel.png"));
41      } catch (IOException ex) {
42          ex.printStackTrace();
43      }
44      lblIcon.setAlignment(Component.CENTER);
45      lblInstruksi = new Label("Pilih Mata Pelajaran");
46      lblInstruksi.getStyle().setBgColor(0xdddeef1);
47      lblInstruksi.setAlignment(Component.CENTER);
48      try {
49          cbMapel = new ComboBox(fetchMapel(nis));
50      } catch (IOException ex) {
51          ex.printStackTrace();
52      }
53  }

```

Gambar 45. Potongan *Script* Halaman Menu Soal

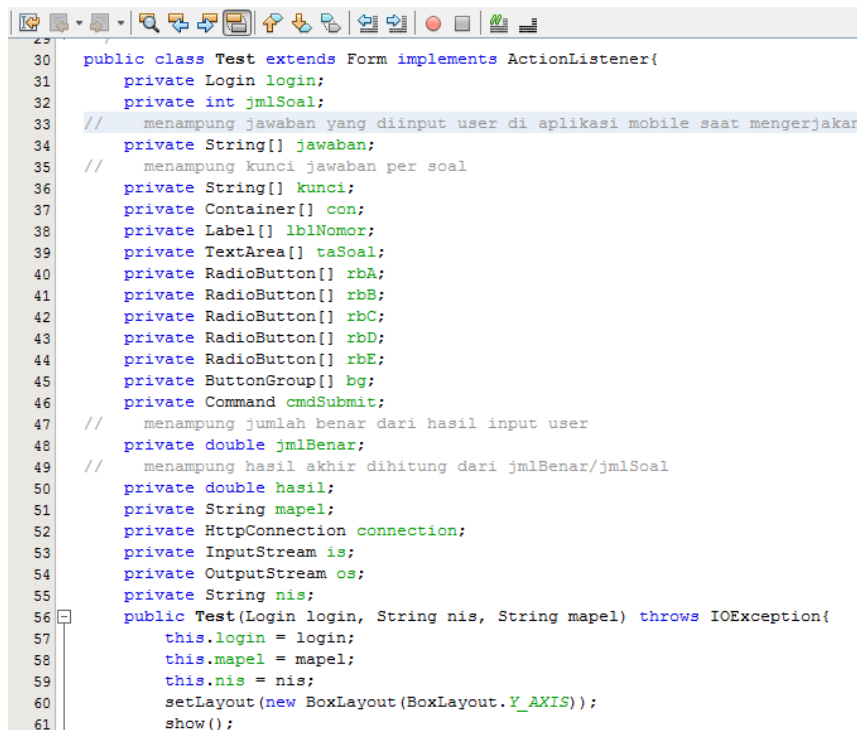
Proses *command* “Pilih” akan melakukan pengiriman *request message* pada *server* untuk mendapatkan *reply message* dari *server* berupa halaman soal. Untuk *command* “Logout” akan melakukan proses keluar dari sistem jika *client/user* tidak melanjutkan proses berjalannya aplikasi Simulasi Tes. Gambar 46 menunjukkan tampilan “Menu” soal pada *emulator Sun Microsystems*.



Gambar 46. Implementasi Halaman Menu Soal

d. Implementasi Halaman Soal

Isi dari halaman soal terdiri atas soal yang diujikan dan *radio button*. Pada proses mengerjakan, *client/user* dihadapkan dengan kursor warna yang apabila saat berpindah soal dan jawaban. *Radio button* juga didukung dengan *command running text* yang memudahkan *client/user* untuk membaca meskipun layar *handphone* terbatas. *Client/user* diharuskan memilih jawaban yang benar yang terdapat pada *radio button*. Berikut penggalan *source code* untuk memunculkan halaman soal:

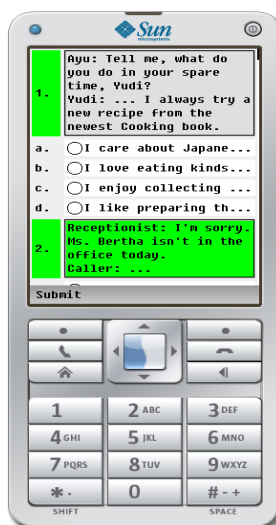


```
30 public class Test extends Form implements ActionListener{
31     private Login login;
32     private int jmlSoal;
33     // menampung jawaban yang diinput user di aplikasi mobile saat mengerjakan
34     private String[] jawaban;
35     // menampung kunci jawaban per soal
36     private String[] kunci;
37     private Container[] con;
38     private Label[] lblNomor;
39     private TextArea[] taSoal;
40     private RadioButton[] rbA;
41     private RadioButton[] rbB;
42     private RadioButton[] rbC;
43     private RadioButton[] rbD;
44     private RadioButton[] rbE;
45     private ButtonGroup[] bg;
46     private Command cmdSubmit;
47     // menampung jumlah benar dari hasil input user
48     private double jmlBenar;
49     // menampung hasil akhir dihitung dari jmlBenar/jmlSoal
50     private double hasil;
51     private String mapel;
52     private HttpConnection connection;
53     private InputStream is;
54     private OutputStream os;
55     private String nis;
56     public Test(Login login, String nis, String mapel) throws IOException{
57         this.login = login;
58         this.mapel = mapel;
59         this.nis = nis;
60         setLayout(new BoxLayout(BoxLayout.Y_AXIS));
61         show();
62     }
63 }
```

Gambar 47. Potongan *Script* Halaman Soal

Command yang digunakan pada halaman soal yaitu *client/user* memilih jawaban pada *radio button*. Soal yang disajikan disertai dengan label berwarna hijau untuk memperjelas *client/user* dalam memahami soal. *Server* akan memroses *request message* berupa jawaban yang divalidasi dengan jawaban

benar pada *database*. *Reply message* dari *server* akan dikonversi menjadi nilai dan ditampilkan pada aplikasi web Simulasi Tes.



Gambar 48. Implementasi Halaman *Input Jawaban*

e. Implementasi Halaman Upload Nilai



Gambar 49. Implementasi Halaman Upload Nilai

File Java yang diperlukan dalam pengembangan aplikasi Simulasi Tes untuk *platform* Java ini di antaranya *DigitalUAN.java*, *Hasil.java*, *Login.java*, *Menu.java*,

SplashScreen.java dan Test.java. File-file ini belum termasuk bawaan Java yang juga berperan dalam proses *build*, *preprocessed*, *prevified*, *preverifysrc*.

4. Implementasi Sistem User

Proses implementasi untuk pengguna menggunakan emulator *Nokia* versi 1.1 dan 2.0 *For Windows* dilakukan untuk memonitor apakah aplikasi Simulasi Tes dapat bekerja sebelum diletakan ke lingkungan yang sebenarnya. Selain itu pengujian dilakukan untuk meninjau tingkat portabilitas dan kompabilitas program dari segi internal. Berikut adalah hasil uji coba menggunakan emulator *Nokia 1.1 For Windows* dengan konfigurasi *device profile* MIDP 2.0 dan resolusi layar: menggunakan pengujian Java dengan versi layar 240x320 *pixel* (HVGA) dan *device profile* MIDP dengan versi 2.1 resolusi layar 320x480 *pixel* (WVGA).



Gambar 50. Pengujian Java 2.0 Bagian *Splash Screen*



Gambar 51. Pengujian Java 2.0 Bagian Halaman *Form Login*



Gambar 52. Pengujian Java 2.1 Bagian *Splash Screen*



Gambar 53. Pengujian Java 2.1 Bagian Halaman *Form Login*



Gambar 54. Pengujian Java 2.1 Bagian Menu Soal



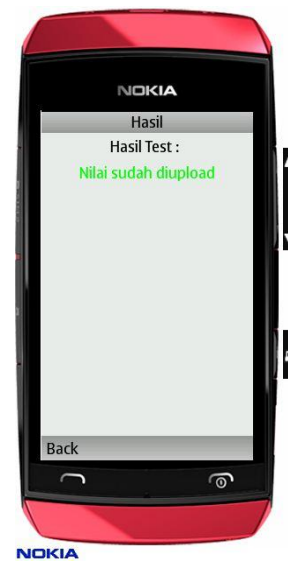
Gambar 55. Pengujian Java 2.1 Bagian Menu Soal



Gambar 56. Pengujian Java 2.1 Bagian Halaman *Input Jawaban*



Gambar 57. Pengujian Java 2.1 Bagian Halaman *Upload Nilai*



Gambar 58. Pengujian Java 2.1 Bagian Halaman *Upload Nilai*



NOKIA

Gambar 59. Pengujian Java 2.0
Bagian Menu Soal



NOKIA

Gambar 60. Pengujian Java 2.0
Bagian Menu Soal



NOKIA

Gambar 61. Pengujian Java 2.0 Bagian Halaman Input Jawaban



NOKIA

Gambar 62. Pengujian Java 2.0
Bagian Halaman *Upload* Nilai



NOKIA

Gambar 63. Pengujian Java 2.0
Bagian Halaman *Upload* Nilai

Hasil pengujian yang dilakukan pada emulator *Nokia 1.1 For Windows* dengan konfigurasi *device profile 2.0* menunjukkan bahwa semua fungsi/fitur simulasi tes yang dikembangkan dapat berjalan secara optimal pada modus *potrait*. Hasil ini menunjukkan bahwa program berjalan sesuai ekspektasi dari segi internal.

Hasil yang sama terlihat juga pada pengujian yang dilakukan dengan *Nokia 2.0 For Windows* dengan konfigurasi *device profile 2.0*. Berikut adalah data hasil pengujian dengan menggunakan emulator *Nokia 2.0 For Windows* dengan konfigurasi *device proile 2.1*.

Hasil pengujian yang dilakukan pada emulator *Nokia 2.0 For Windows* dengan konfigurasi *device proile 2.1* menunjukkan bahwa semua fungsi/fitur simulasi tes yang dikembangkan dapat berjalan secara optimal pada modus *potrait*. Hasil ini menunjukkan bahwa program berjalan sesuai ekspektasi.

D. Pengujian Perangkat Lunak

1. *White Box Testing*

Uji *white box testing* merupakan proses pengujian perangkat lunak dari sisi desain dan kode program. Program diuji apakah mampu menghasilkan fungsi-fungsi yang sesuai dengan kebutuhan dan tidak mengalami kesalahan dari sisi program untuk mengetahui kelayakan aspek *correctness* sesuai kaidah *software quality*. Dalam pengembangan produk *software* ini, pengujian *white box* menggunakan mekanisme perhitungan *Kilo Line Of Code*. Untuk itu, terlebih dahulu dilakukan perhitungan jumlah baris kode untuk masing-masing *file* yang berisikan kode program. *File* beserta baris kode yang dimiliki dalam tabel 45 berikut ini:

Tabel 45. Pengujian Aspek *Correctness*

Nama File	Jumlah Baris Kode
Folder/src	
DigitalUAN.java	33
SplashScreen.java	44
Login.java	145
Menu.java	135
Test.java	228
Folder/nbproject	
Build-impl.xml	1430
Project.xml	10
Private.xml	4
Folder/views	
Header.php	4
Footer.php	6
Template.php	39
Welcome_message.php	88
Folder/DigitalUAN	
daftarMapel.php	22
dataTes.php	6
Login.php	11
Upload.php	17
Folder/content	
Daftar_siswa.php	9
Data_pembahasan_pengayaan.php	18
Data_pembahasan_utama.php	20
dataPengayaan.php	63
Home_guest.php	10
Home_guru.php	6
Home_siswa.php	6
Nilai_guru.php	9
Nilai_siswa.php	9

Pembahasan_pengayaan.php	1
Pembahasan_utama.php	1
Pengayaan.php	1
Folder/sidebar	
Sidebar_guest.php	32
Sidebar_guru.php	21
Sidebar_siswa.php	46
Total Baris Kode	2474 Line Of Code
Total KLOC	2.474 KLOC

Berdasarkan perhitungan baris kode pada *source code* aplikasi Simulasi Tes berbasis *mobile application* menggunakan bahasa pemrograman *Java Micro Edition*, *PHP* dan *MySQL* didapatkan nilai 2474 LOC atau 2.474 KLOC. Nilai tersebut kemudian dimasukkan ke dalam model perhitungan KLOC yang ditujukan pada tabel 46 berikut:

Tabel 46. Metode Perhitungan *Correctness* (Pressman, 2010:709)

Metode	Rumus
Walston-Felix Model	$E = 5.2 \times (\text{KLOC})^{0.91}$
Bailey-Basili Model	$E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$
Boehm simple Model	$E = 3.2 \times (\text{KLOC})^{1.05}$
Doty model for KLOC > 9 (untuk lebih dari 9000 baris kode)	$E = 5.288 \times (\text{KLOC})^{1.047}$

*) Keterangan: Model Doty tidak digunakan karena total baris kurang dari 9000 baris.

Berikut ini adalah hasil perhitungan menggunakan metode Waltson-Felix, Bailey-Basili dan Boehm untuk pengujian *correctness*:

Tabel 47. Pengujian Aspek *Correctness*

Model Waltson-Felix : $E = 5.2 \times (2.474)^{0.91}$ E = 11.8
Model Bailey-Basili : $E = 5.5 + 0.73 \times (2.474)^{1.16}$ E = 7.58
Model Boehm : $E = 3.2 \times (2.474)^{1.05}$ E = 8.29

Nilai densitas *error* yang diperoleh adalah 11.8, 7.58 dan 8.29. dengan ukuran proyek kurang dari 16.000 baris kode, maka nilai densitas *error* tersebut berada pada rentang 0–40 *error* per *KLOC* sebagaimana dikemukakan oleh Steve McConnell (2004: 5). Oleh karena itu, dapat disimpulkan bahwa Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* telah memenuhi kaidah *software quality* untuk aspek *correctness*.

2. **Black Box Testing**

Uji *Black Box Testing* merupakan proses pengujian perangkat lunak dari segi spesifikasi performa program. Pengujian ini dilakukan tanpa menguji desain dan kode program. Pengujian ini digunakan untuk mengetahui apakah konsep dan perancangan perangkat lunak sudah sesuai dengan spesifikasi yang diperlukan.

Pada tahapan pengujian *black box* dilakukan pengujian sesuai dengan fungsionalitas perangkat lunak pada desain *use-case*. Peneliti membagi pengujian menjadi 11 (sebelas) bagian. Setiap bagian diuji sesuai dengan skenario pada desain *use case* pada tahapan desain sistem. Setelah mengalami pengujian didapatkan hasil uji *black-box testing* seperti berikut ini:

Tabel 48. Uji *Black-box Testing* Menjalankan Aplikasi

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. Menjalankan Aplikasi.		
	2. Menjalankan splash screen dan melakukan inisialisasi <i>database</i> . 3. Setelah selesai inisialisasi <i>database</i> kemudian menuju ke <i>form login</i> .	Sesuai

Tabel 49. Uji *Black-box Login* Aplikasi

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> mengisi <i>username</i> dan <i>password</i> pada <i>form login</i> .		
	2. Melakukan inisialisasi <i>database</i> . 3. Setelah selesai inisialisasi <i>database</i> kemudian menuju ke pilihan soal.	Sesuai

Tabel 50. Uji *Black-box Login* Aplikasi

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> memilih pilihan mata pelajaran yang akan diujikan.		
	2. Melakukan inisialisasi <i>database</i> . 3. Setelah selesai inisialisasi <i>database</i> kemudian menuju ke menu <i>test</i> untuk mengerjakan soal.	Sesuai

Tabel 51. Uji *Black-box Menu Test*

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> mengerjakan soal dengan memilih pilihan jawaban.		
	2. Melakukan pengiriman data nilai ke <i>server</i> . 3. Setelah selesai melakukan pengiriman data ke <i>server</i> kemudian menuju ke menu <i>upload</i> nilai.	Sesuai

Tabel 52. Uji *Black-box* Menu Test

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> memilih pilihan pesan konfirmasi dari sistem.		
	2. kembali menuju menu pilihan soal. 3. memilih pilihan mata pelajaran yang belum dikerjakan.	Sesuai

Tabel 53. Uji *Black-box* Menu *Login* Aplikasi Web

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> membuka browser dan membuka alamat rizam.fastacenter.com. 2. <i>User</i> mengisi id <i>username</i> dan <i>password</i> pada <i>form login</i> .		
	3. Melakukan insialisasi <i>database</i> . 4. Setelah melakukan insialisasi <i>database</i> kemudian menuju ke <i>home page</i> .	Sesuai

Tabel 54. Uji *Black-box* Menu Nilai

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> memilih fitur "Nilai".		
	2. Melakukan insialisasi <i>database</i> . 3. Setelah melakukan insialisasi <i>database</i> kemudian muncul tabel nilai <i>user</i> .	Sesuai

Tabel 55. Uji *Black-box* Menu Pembahasan Soal

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> memilih fitur "Pembahasan Soal".		
	2. Melakukan insialisasi <i>database</i> . 3. Setelah melakukan insialisasi <i>database</i> kemudian muncul pembahasan soal.	Sesuai

Tabel 56. Uji *Black-box* Menu Pengayaan

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> memilih fitur "Pengayaan".		
	2. Melakukan insialisasi <i>database</i> . 3. Setelah melakukan insialisasi <i>database</i> kemudian muncul soal sebagai pengayaan nilai.	Sesuai

Tabel 57. Uji *Black-box* Menu *Logout*

Aksi Aktor	Reaksi Sistem	Hasil Pengujian
Skenario Normal		
1. <i>User</i> melakukan <i>Logout</i> dari aplikasi		
	2. Melakukan insialisasi <i>database</i> . 3. Setelah melakukan insialisasi <i>database</i> kemudian muncul soal sebagai pengayaan nilai.	Sesuai

E. Alpha Testing

Proses uji *alpha* merupakan tahapan pengujian yang dilakukan oleh pengguna yang dipilih (*selected users*). Pengguna terpilih tersebut nantinya akan menguji perangkat lunak untuk mengetahui kualitas perangkat lunak pada aspek

reliability, efficiency, integrity, maintainability, flexibility, testability, reusability dan *interoperability*. Pengujian *alpha* menurut Pressman merupakan tahapan pengujian dari sisi pengembang. Pengujian dilihat dari perspektif dan lingkungan pengembang. Pengujian dilakukan untuk mendapatkan *error* permasalahan penggunaan dari sisi pengembang (Pressman, 2001: 496).

Kuesioner pada uji *alpha* ini dibuat berdasarkan sub-karakteristik dari faktor kualitas perangkat lunak menurut *McCall's quality factors* pada faktor *reliability, efficiency, integrity, maintainability, flexibility, testability, reusability* dan *interoperability*. Kuesioner tersebut sebelum digunakan telah lolos dalam validasi angket oleh 3 (tiga) *validator* yang memiliki spesifikasi atau keahlian mengenai antar muka, pemrograman, basis data dan rekayasa perangkat lunak.

Dalam tahapan uji *alpha* perangkat lunak diuji oleh 5 ahli dalam bidang *software engineering*. Pengujian mengacu pada karakteristik *reliability, efficiency, integrity, maintainability, flexibility, testability, reusability* dan *interoperability*. Hasil pengujian menunjukkan semua spesifikasi pada pengujian *alpha* telah menunjukkan hasil kerja yang benar. Hal ini di buktikan dengan tabel 58 pengujian *alpha* yang mendapatkan hasil pengujian yang telah sesuai dan mendapatkan kesimpulan perangkat lunak dapat bekerja secara baik. Berikut adalah hasil proses pengumpulan data pada uji *alpha* didapat data seperti berikut:

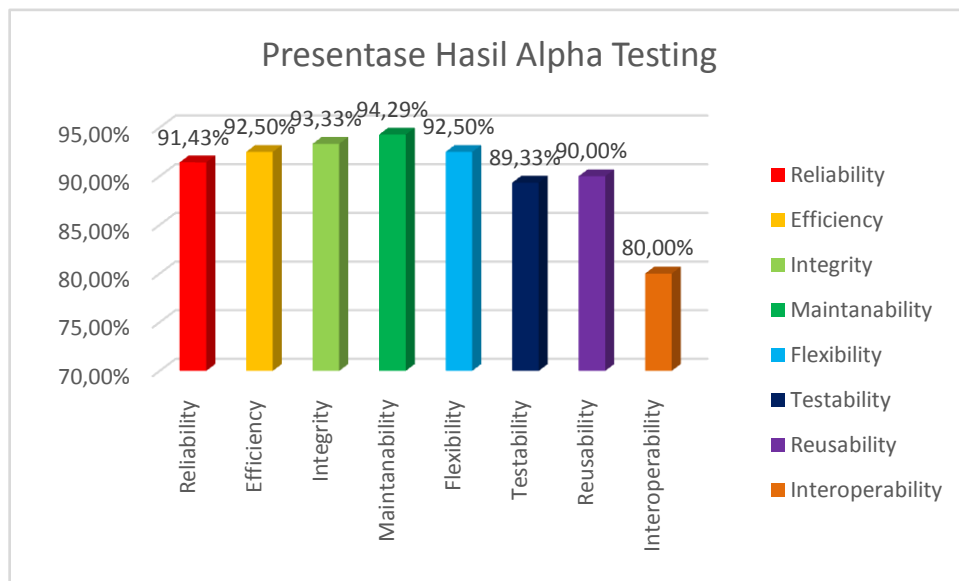
Tabel 58. Hasil Perhitungan *Alpha Testing*

Responden	RE	EF	IN	MN	FL	TS	RS	IT
A	14	8	18	14	16	12	10	5
B	12	6	16	12	15	13	11	5
C	14	8	15	13	13	14	12	6
D	13	7	17	13	14	13	10	6
E	11	8	18	14	16	15	11	6
Total Skor	64	37	84	66	74	67	54	28
Skor Maksimum	70	40	90	70	80	75	60	35
Rerata Item	12.8	7.4	16.8	13.2	14.8	13.4	10.8	5.6
Persentase (%)	91.43	92.50	93.33	94.29	92.50	89.33	90.00	80.00

Keterangan :

RE : <i>Reliability</i>	FL : <i>Flexibility</i>
EF : <i>Efficiency</i>	TS : <i>Testability</i>
IN : <i>Integrity</i>	RS : <i>Reusability</i>
MN : <i>Maintanability</i>	IT : <i>Interoperability</i>

Selain memberikan pengujian dan penilaian terhadap unjuk kerja perangkat lunak, ahli perangkat lunak juga memberikan beberapa saran dalam pengembangan perangkat lunak tersebut. Saran dalam pengembangan perangkat lunak agar ditambahkan kolom *radiobutton* yang awal pengembangan aplikasi berjumlah 4 kolom dikembangkan menjadi 5 kolom. Saran tersebut digunakan pada tahapan revisi perangkat lunak.



Gambar 67. Diagram Persentase Kualitas Perangkat Lunak

(1) Rumus perhitungan persentase skor ditulis dengan rumus berikut:

$$\text{Persentase Kelayakan} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

(2) Hasil perhitungan yang didapatkan :

$$\text{Persentase Kelayakan (\%)} = \frac{474}{520} \times 100\%$$

$$\text{Persentase Kelayakan} = 91.15\%$$

Hasil perhitungan yang digunakan untuk mengetahui kualitas perangkat lunak dari segi *reliability*, *efficiency*, *integrity*, *maintainability*, *flexibility*, *testability*, *reusability* dan *interoperability* pada perangkat lunak masuk dalam kategori “Sangat Baik”. Sedangkan untuk masing-masing faktor tingkat kelayakannya adalah sebagai berikut ini:

Tabel 59. Persentase Kelayakan Tiap Faktor

No.	Faktor	Persentase	Tingkat Kelayakan
1.	Reliability	91.43%	Sangat Baik
2.	Efficiency	92.50%	Sangat Baik
3.	Integrity	93.33%	Sangat Baik
4.	Maintainability	94.29%	Sangat Baik
5.	Flexibility	92.50%	Sangat Baik
6.	Testability	89.33%	Baik
7.	Reusability	90.00%	Sangat Baik
8.	Interoperability	80.00%	Baik

Dari hasil pengolahan data pengujian yang dilakukan oleh 5 ahli rekayasa perangkat lunak dari beberapa perusahaan pengembang perangkat lunak secara keseluruhan adalah 91.15%. dari hasil keseluruhan tersebut maka dapat disimpulkan kualitas perangkat lunak tersebut memiliki kategori “Sangat Baik”.

F. *Beta Testing*

1. Pengujian Aspek *Usability*

Pengujian *beta* merupakan pengujian pada sisi pengguna. Pengujian *beta* dilakukan untuk mendapatkan usulan dari pengguna akhir sebelum persiapan rilis produk akhir (Pressman, 2001: 496). Pada pengujian *beta* tahapan pengujian dilakukan oleh sampel pengguna kepada 31 siswa kelas 3 AV 1 di SMK Negeri 3 Yogyakarta. Hasil pengujian *usability* dapat ditunjukkan pada tabel 46. Berikut adalah hasil proses pengumpulan data pada uji *beta* didapat data pada tabel 59 berikut:

Tabel 60. Hasil Pengujian *Usability*

Responden	Pernyataan																			Total
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
14419	3	7	4	1	1	3	3	3	7	3	4	4	4	4	3	3	4	3	4	68
14420	4	5	4	1	2	3	2	5	4	4	5	3	4	4	5	6	5	5	6	145
14421	7	7	7	7	7	6	6	6	6	6	6	6	6	6	5	7	5	6	6	118
14422	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
14423	5	6	3	3	3	5	6	3	3	6	5	3	5	5	5	6	6	6	3	87
14424	6	6	7	5	7	7	6	6	6	7	7	6	7	7	6	6	6	5	6	119
14425	4	3	5	3	3	5	5	5	6	5	6	6	6	4	3	4	3	2	5	83
14426	7	6	7	7	6	6	7	6	3	7	7	6	6	7	4	5	5	7	6	115
14427	7	6	7	7	6	7	7	6	4	7	7	7	6	7	5	5	5	6	7	119
14428	4	2	3	3	4	4	4	3	5	4	5	5	5	6	4	5	4	4	7	81
14429	5	6	6	5	5	5	5	5	3	4	5	4	5	5	5	5	5	5	6	94
14430	7	7	7	7	7	7	7	7	6	7	7	7	7	7	4	5	6	5	6	123
14431	6	6	4	6	7	7	5	5	6	4	3	3	5	4	3	6	5	5	6	96
14432	7	6	6	6	6	6	7	7	7	6	7	7	6	6	6	7	7	6	7	123
14434	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
14435	7	6	6	5	5	5	5	5	3	4	6	5	6	7	5	5	5	5	5	100
14436	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
14437	4	2	3	3	7	7	7	7	5	7	5	5	4	4	4	5	6	5	4	94
14438	4	5	4	1	2	3	2	5	4	4	5	3	4	4	5	6	5	5	6	77
14439	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
14440	6	6	5	5	5	5	4	5	6	7	5	4	4	5	6	5	4	5	4	96
14443	7	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	131
14444	7	6	6	4	5	6	7	6	7	6	6	6	7	6	7	7	7	6	7	119
14445	6	7	5	7	6	7	7	7	7	6	7	7	6	7	6	7	7	7	7	126
14446	7	7	5	5	6	7	7	6	4	5	7	6	7	7	6	6	7	6	7	118
14447	6	5	5	2	3	5	3	5	4	3	3	5	4	6	3	3	6	4	6	81
14448	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
14449	5	5	3	4	4	6	6	5	5	5	4	4	5	5	5	5	4	5	5	90
14451	4	2	3	3	4	4	4	3	5	4	5	5	5	6	4	5	4	4	7	81
14452	7	7	5	5	4	7	7	6	7	7	6	6	6	7	5	6	7	6	6	117
14453	6	7	6	7	5	5	7	5	6	5	6	5	6	7	5	5	5	5	5	108

Setelah melakukan pengujian aspek *usability*, kemudian melakukan perhitungan nilai koefisien *alpha cronbach* menggunakan perangkat lunak program *SPSS Statistics 17 for Windows*. Hasil perhitungan ditunjukkan pada tabel 61 sebagai berikut:

Tabel 61. Hasil Perhitungan *Alpha Cronbach*

Case Processing Summary			
		N	%
Cases	Valid	31	100.0
	Excluded ^a	0	.0
	Total	31	100.0
a. Listwise deletion based on all variables in the procedure.			

Reliability Statistics	
Cronbach's Alpha	N of Items
.965	19

Untuk melakukan perbandingan dan penghitungan deskriptif persentase dari data hasil pengujian *beta* digunakan rumus:

(1) Rumus perhitungan persentase skor ditulis dengan rumus berikut:

$$\text{Persentase Kelayakan} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

(2) Hasil perhitungan yang didapatkan :

$$\text{Persentase Kelayakan (\%)} = \frac{3379}{4123} \times 100\%$$

$$\text{Persentase Kelayakan} = 82.00\%$$

Dari hasil pengolahan data pengujian yang dilakukan oleh pengguna menunjukkan bahwa aplikasi Simulasi Tes telah memenuhi aspek *usability* dengan nilai *alpha cronbach* 0.965 dan persentase 82.00%. Dari hasil keseluruhan tersebut maka dapat disimpulkan kualitas perangkat lunak tersebut memiliki kategori “**Sangat Baik**”.

BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan pengembangan yang dilakukan dalam mengembangkan perangkat lunak Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* maka peneliti mengambil kesimpulan:

1. Pengembangan aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL* dilakukan dengan beberapa tahap yaitu analisis (analisis kebutuhan, analisis *hardware*, analisis *software*), desain (desain struktur program dan desain *interface*), pengembangan (perancangan desain *interface* dan pembuatan program), pengujian (*white-box testing*, *black-box testing*, *alpha testing*, *beta testing* berdasarkan Mccall's *Software Quality*). Aplikasi Simulasi Tes Berbasis *Mobile Application* dikembangkan menggunakan *software* NetBeans IDE 7.0.1 yang merupakan *software* pencipta aplikasi berbasis *java programming*.
2. Hasil pengujian kualitas aplikasi Simulasi Tes Berbasis *Mobile Application* yang meliputi aspek *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintanability*, *flexibility*, *testability*, *reusability*, *interoperability* yaitu sebagai berikut:
 - a. Hasil pengujian kuailitas aspek *correctness*, aplikasi Simulasi Tes Berbasis *Mobile Application* menggunakan perhitungan *KLOC* (*Kilos Lines of Code*) memiliki **2474** baris kode program dan hasil pengukuran

densitas *error* menurut perhitungan model Watson - Felix nilai densitas *error* sebesar **11.8**, perhitungan model Bailey - Basili nilai densitas *error* **7.58** dan perhitungan model Boehem nilai densitas *error* sebesar **8.29**. Untuk rentang ukuran proyek kurang dari 16.000 baris kode, nilai densitas *error* yang diperoleh tersebut berada dalam rentang 0 – 40 *error* per *KLOC*. Hal ini menunjukkan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP dan MySQL* telah memenuhi aspek *correctness* menurut kaidah *software quality*.

- b. Hasil pengujian kualitas aspek ***reliability*** oleh 5 orang ahli rekayasa perangkat lunak, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik ***accuracy, consistency, error tolerance, modularity*** dan ***simplicity***. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **91.43%**.
- c. Hasil pengujian aspek ***efficiency*** oleh 5 orang ahli rekayasa perangkat lunak, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik ***conciseness, execution efficiency*** dan ***operability***. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **92.50%**.
- d. Hasil pengujian aspek ***integrity*** oleh 5 orang ahli, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik ***auditability, instrumentation*** dan ***security***. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **93.33%**.
- e. Hasil pengujian aspek ***maintanability*** oleh 5 orang ahli, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik ***conciseness***,

consistency, instrumentation, modularity, self-documentation dan *simplicity*. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **95.29%**.

- f. Hasil pengujian aspek *flexibility* oleh 5 orang ahli rekayasa perangkat lunak, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik *conciseness, consistency, expandability, generality, modularity, self-documentation* dan *simplicity*. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **92.50%**.
- g. Hasil pengujian aspek *testability* oleh 5 orang ahli, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik *auditability, instrumentation, modularity, self-documentation* dan *simplicity*. Sehingga kualitas perangkat lunak mempunyai skala “**Baik**” dengan presentase **89.33%**.
- h. Hasil pengujian aspek *reusability* oleh 5 orang ahli, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik *generality, hardware independence, modularity, self-documentation* dan *software system independence*. Sehingga kualitas perangkat lunak mempunyai skala “**Sangat Baik**” dengan presentase **90.00%**.
- i. Hasil pengujian aspek *interoperability* oleh 5 orang ahli, menunjukkan bahwa aplikasi dapat berjalan sesuai uji sub-karakteristik *communication commonality, data commonality, expandability* dan *modularity*. Sehingga kualitas perangkat lunak mempunyai skala “**Baik**” dengan presentase **80.00%**.

- j. Hasil pengujian kualitas aspek **usability** oleh 31 siswa kelas XII AV 1 SMK N 3 Yogyakarta diperoleh nilai alpha cronbach **0,965** dengan skala kualitas “**Sangat Baik**”. Dengan demikian, dapat disimpulkan bahwa Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP dan MySQL* telah memenuhi aspek *usability* menurut kaidah *software quality*.

B. Saran

Dalam pengembangan perangkat lunak yang tentunya masih terdapat kekurangan dan hal-hal yang masih perlu dikaji dan dikembangkan kembali. Peneliti memiliki pemikiran dan saran untuk pengembangan ke depan antara lain:

1. Bagi Siswa

- a. Siswa dapat menggunakan aplikasi Simulasi Tes Berbasis *Mobile Application* sebagai evaluasi belajar.
- b. Siswa dapat menggunakan aplikasi Simulasi Tes Berbasis *Mobile Application* untuk membantu belajar siswa sebelum menghadapi Ujian Sekolah atau Ujian Nasional.

2. Bagi Sekolah

- a. Pihak sekolah perlu mengupayakan peningkatan kemampuan guru dalam bidang pengembangan media pembelajaran yang berbasis teknologi, sehingga dapat menunjang proses belajar siswa.
- b. Perlu adanya dukungan dari guru untuk menyampaikan tujuan evaluasi pembelajaran berbasis *mobile application* kepada siswa.

3. Bagi Pengembang

- a. Materi yang ada dalam aplikasi lebih kompleks dan mengacu kurikulum yang relevan, sehingga siswa dapat mengikuti pembelajaran sesuai kurikulum.
- b. Mengembangkan aplikasi yang dapat berjalan dalam *platform* Android dan menambah jumlah mata pelajaran yang diujikan.

DAFTAR PUSTAKA

- Agarwal, B. B., Tayal, S. P., & Gupta, M. (2010). *Software engineering and testing*. Massachusetts: Jones and Bartlett Publishers.
- Ahfas, I. (2012). *Rancang Bangun Simulasi Tes Online Calon Pegawai Negeri Sipil (CPNS)*. Kudus: Universitas Muria Kudus.
- Anastasi, A. (1983). *Psychological testing: Basic concepts and common misconceptions*. Diakses dari <http://www.nytimes.com/2001/05/16/anne-anastasi-the-test-guru-of-psychology-is-dead-at-92.html>. Pada tanggal 12 Februari 2013, Jam 22.00 WIB
- Anhar. (2010). *Panduan Menguasai PHP & MySql Secara Otodidak*. Jakarta: Agromedia Pustaka.
- Aryoyudanta, B. (2012). *Aplikasi Pemetaan Rute dan Halte Trans Jogja Berbasis Geolocation pada Smartphone Android*. Yogyakarta: Universitas Negeri Yogyakarta.
- Baskara, E. (2012). *Aplikasi Kamus Istilah Komputer Dan Informatika Berbasis J2ME (Java 2 Micro Edition)*. Yogyakarta: Universitas Negeri Yogyakarta
- Bloom, B. S., Englehart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *The taxonomy of educational objectives: affective and cognitive domains*. New York: Longman
- Borg, W. R., & Gall, M. D. (1996). *Educational research: An introduction (6th ed.)*. England: Longman Publishing.
- Cronbach, L. J. (1970). *Essential of Psychological Testing*. New York: Harper and Row.
- Depdiknas. (2005). *Kumpulan Pembelajaran/Pendampingan*. Jakarta: Balai Pustaka.
- Djamarah, S. B. (2008). *Psikologi Belajar*. Jakarta: Rineka Cipta.
- Djati, B. S. (2007). *Simulasi, Teori dan Aplikasinya*. Yogyakarta: Penerbit Andi.
- Georgiev, & Tsvetozar. (2004). *M-Learning – a New Stage of E-Learning* (Online), conference on computer systems and technologies. Diakses dari <http://ecet.ecs.ru.acad.bg/cst04/docs/siv/428.pdf>, Pada 30 Desember 2013., Jam 07.30 WIB

- Hasan, I. (2010). *Analisis Data Penelitian Dengan Statistik*. Jakarta: Bumi Aksara.
- Hasan, M. I. (2002). *Pokok-Pokok Materi Pengambilan Keputusan*. Jakarta: Ghalia Indonesia.
- Hazmin, G. (2013). Aplikasi Simulasi Tes Penerimaan Calon Pegawai Negeri Sipil (CPNS) Pada Pengadilan Tinggi Agama Yogyakarta Menggunakan App Inventor. Diakses dari repository.amikom.ac.id/files/Publikasi_09.11.3285.pdf. Pada tanggal 20 April 2014. Jam 01.00 WIB
- Hobban, G., & Ferry, B. (2006). *Teaching science concepts in higher education classes with slow motion animation (slowmation)*. Diakses dari <http://www.editlib.org/p/26211/>. Pada tanggal 20 April 2014. Jam 11.50 WIB.
- IEEE. (1990). *IEEE Standard Glossary of Software Engineering Technology*. New York: IEEE.
- JENI. (2011). *Pengembangan Aplikasi Mobile*. Diakses dari [http://poss.ipb.ac.id/files/JENI-J2ME-Bab01 Pengembangan%20Aplikasi%20Mobile.pdf](http://poss.ipb.ac.id/files/JENI-J2ME-Bab01%20Pengembangan%20Aplikasi%20Mobile.pdf). Pada tanggal 22 Februari 2011, Jam 02.00 WIB
- Kadir, A. (2008). *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Andi.
- Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel Psychology*, Vol 28, Iss. 4, pp. 563-575.
- Lewis, J. R. (1995). *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. International Journal of Human-Computer Interaction, 7, Hlm. 57-78.
- Lincke, R. (2007). *Validation of a Standard- and Metric-Based Software Quality Model*. Sweden: Växjö University.
- Marsh, E. J. (2005). The positive and negative consequences of multiple-choice testing. *Journal Of Experimental Psychology: Learning, Memory, And Cognition*, Vol. 31, No. 5, pp. 1155–1159.
- McConnell, S. (2004). *Code Complete, A Practical Handbook of Software Construction : 2nd Edition*. Microsoft Press.
- Meulen, M. V. (2000). *Definitions for hardware and software safety engineers*. London: Springer-Verlag.

- Nugraha, A. (2011). Pengembangan dan Implementasi Mobile Learning Berbasis J2ME Untuk Mata Pelajaran Keterampilan Komputer dan Pengelolaan Informasi. Seminar. Bandung: Pendidikan Ilmu Komputer UPI.
- O'Malley, C, et al. 2003. *Guidelines For Learning/Teaching/Tutoring in a Mobile Environment* (Online).
Diakses dari <http://www.mobilelearn.org/download/results/guidelines.pdf>.
pada 29 Agustus 2013, Jam 07.30 WIB
- Pressman, R. S. (2001). *Rekayasa Perangkat Lunak : Pendekatan Praktisi*. Yogyakarta: Andi.
- Pressman, R. S. (2001). *Software Engineering A Practitioner's Approach*. New York: McGraw-Hill.
- Pressman, R. S. (2000). *Software Engineering: A Practitioner's Approach (European Adaptation), Ch. 19, (5th Ed)*. New York: McGraw-Hill.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach, (7th Ed)*. New York: McGraw-Hill Companies.
- repository.usu.ac.id/bitstream/.../4/Chapter%20II.pdf. (2012). Medan: Universitas Sumatra Utara.
- Riduwan. (2011). *Belajar Mudah Penelitian untuk Guru-Karyawan dan Peneliti*. Bandung: Alfa Beta.
- Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Massachusetts: Addison Wesley Longman.
- Saepudin, A. (2003). Penerapan Teknologi Informasi dalam Pendidikan Masyarakat. *Jurnal Teknodik* Edisi No.12/VII/Oktober/2003.
- Setemen, K. (2010). Pengembangan Evaluasi Pembelajaran Online. *Pengembangan Evaluasi Pembelajaran Online*. Diakses dari <http://ejournal.undiksha.ac.id>. Pada tanggal 20 Februari 2013, Jam 10.00 WIB.
- Shalahuddin, M., & Rosa, A. S. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- Slamet, D. (1988). *Evaluasi Pendidikan*. Jakarta: PT. Bina Aksara.
- Soesilo, M., Hartati, G. S., & Suharto, B. H. (2005). *Java 2 SE dengan JBuilder*. Yogyakarta: Andi.
- Sonjaya, I., Kuspriyanto, & Aciek Ida Wuryandari. (2010). *Perancangan Sistem Pembelajaran Berbasis Web Dengan Menggunakan Pendekatan Model Cooperative Learning (CI)*. Jurnal dan Penelitian. Hlm. 1-10.
- Sugiyono. (2009). *Statistika Untuk Penelitian*. Bandung : Alfabeta.

- Sukmawati, M. (2009). *Rancang Bangun Aplikasi Simulasi TOEFL Berbasis Android*. Palembang: STMIK GI MDP.
- Susanto, J. (2011). Aplikasi Route Bus Transjogja dan Informasi Wisata Kota Yogyakarta Berbasis J2ME. Diakses dari http://eprints.uny.ac.id/6329/1/juni_susanto.pdf. Pada tanggal 22 Februari 2013, Jam 20.00 WIB
- Sutarno. (2010). Online Interaktif Multimedia. Diakses dari <http://repository.upi.edu/9881/>. Pada tanggal 20 Februari 2013, Jam 09.35 WIB
- Taptozani, M. A. (2009). Pembuatan Aplikasi Mobile Pemesanan Makanan Dan Minuman Berbasis J2me Dan Bluetooth. Proyek Akhir. Surabaya. Institut Teknologi Sepuluh November.
- Valade, J. (2004). *PHP & MySQL For Dummies (2nd ed)*. Indiana: Wiley Publishing.
- Westen, D., & Rosenthal, R. (2003). Quantifying Construct Validity: Two Simple Measures. *Journal of Personality and Social Psychology*, Vol 84, No 3, pp. 608-618.
- Widoyoko, E. P. (2012). *Teknik Penyusunan Instrumen Penelitian*. Yogyakarta: Pustaka Pelajar.
- Wijaya, W. G. (2010). Aplikasi Tes Online Dengan Teknik Pemrograman Client Server. Diakses dari <https://www.scribd.com/doc/94539268/Jurnal-Aplikasi-Tes-Online>. Pada tanggal 23 Februari 2013, Jam 21.00 WIB
- Yadav, S. C., & Singh, S. K. (2009). *An introduction to client/server computing*. New Delhi: New Age International.

LAMPIRAN

Lampiran 1. Materi Soal Bahasa Inggris Aplikasi Simulasi Tes

Soal Simulasi Tes Bahasa Inggris

1. Dewi : Hello, would you like to come to my house tonight? We will have a birthday party for my youngest sister.
Dewa : Yes, Sure....
 - a. I'd love to
 - b. I don't think so
 - c. I have a lot of to do
 - d. Would you like to have dinner with me?
 - e. I don't anything
2. Aryo : Good morning. This is Aryo Calling. May I speak to Mr. Agus Please?
Receptionist : ... I'll try to put you through to Mr. Agus at his office.
 - a. It's all right
 - b. Hold the line, please
 - c. Who's speaking, please?
 - d. Sorry to keep you waiting.
 - e. Oh, not me.
3. Diane : What do you like most in this city?
Mary : Well, There's so much to see.
 - a. I enjoy going shopping quite a lot.
 - b. I'm very fond of the food.
 - c. I love the steak very much.
 - d. I really like hot weather.
 - e. I will do it.
4. Ms. Dilla: Anything to do tonight, Sari?
Ms. Sari : I'm afraid yes, ... a chef are Sheraton Hotel?
 - a. I had to see
 - b. I'll go home
 - c. You will accompany.
 - d. You should accompany.
 - e. I'm going to meet
5. Stranger : Could you tell me how to get to the bank?
You :
 - a. It's next to the shopping center.
 - b. Go straight ahead and turn left.
 - c. It's around the corner from the station.
 - d. The bank is located across from the supermarket.
 - e. It's around the corner from the Bank.

6. Mrs. Thomas : I've just witnessed an accident. It was a hit and run.
 Police Officer : Can you describe the car?
 Mrs. Thomas : Well,... I think.
- It's green with buttons down the front.
 - It's a two storey one with air-conditioning
 - It's a small blue Japanese one with two doors.
 - It's an expensive one with a wheel a the back.
 - It's an cheap one with a wheel a the back.
7. Lula : Excuse me. May you see Mr. Sroyo?
 Kamal : I'm sorry madam. He A client at present.
- Can contact.
 - Was visting.
 - Was gone.
 - Had a chat with.
 - Is talking to.
8. Manager : Did our company get more profit of the sales this year?
 Supervisor : I'm afraid not. Sir. But we could have been able to promote the
 products...
- If we had hired more sales representatives.
 - If we hadn't produced mre products than ever.
 - If people hadn't bought other products but ours.
 - If people had been familiar with our.
 - If people has been familiar with our.
9. Amin : I'm sorry for being late. My car broke down on my way here.
 Mr. Budi :
 Amin : Thanks for your advice. I'll do that.
- You should call taxi.
 - My driver can help you.
 - I think I will send you a driver.
 - You'd better have it checked regulary.
 - You must be patient.
10. Teacher : Is Bakti present today?
 Lodui : No. Sir. Yesterday he was present, but he looked so pale....
- He must be fine.
 - He might be sick.
 - He may be all right.
 - He is studying English.
 - He was sick.
11. Applicant : Are there any other requirements for this position?
 A

- Staff : Yes, you have to be bilingual.
B
- Applicant : Well, I may speak English and a little German.
C D E
12. Dora : Should I work overtime to meet the deadline, Sir?
A B C
- Mr. Surya : No, you not have to, unless you can finish it on time.
D E
13. My opinion that is medicine makes us healthy.
A B C D E
14. A week ago Martha was knocking down by a car. Then she was taken to hospital.
A B
C
Fortunately, she wasn't badly injured.
D E
15. Assistant : Here's the report you asked for. Is there anything else I should do?
A B
- Accountant : Yes, could you please to help me put this file back before leaving?
C D
E
16. Rekha : Who do you think will be the winner in the next 100 meter Olympics race?
A
17. Rian : I can surely say that American sprinters still have the most good chance for it.
B C D E
18. Mr. black: So, how was the seminar?
A
Cole : Not good. Some of the speaker's remarks were really annoyed.
B C D E

19. Tery : Going home, Candy? Should you go home with me?
A B
Candy : Thank you very much. That's very kind of you.
C
Tery : Never mind. I will pass your apartment anyway.
D E
20. The Owner : Could you write down how many gas stoves there are?
A
The Shop-assistant: Sure. I've counted that there are only one gas stove, but
B C
There are still twenty electric stoves.
D E
21. Anita : I'm calling about tomorrow's meeting.
A
Ressa : Yes. What's up?
B
Anita : Will we postpone it? The manager won't be here until next week.
C D E

Taylor, Inc.
694 Rockstar Lane
Durham, NC 27708

Dear Durham Human Resources Director:

I just read an artikcle in *New and Observer* about Taylor's new computer center just north of Durham. I would like to apply for a position as an entry-level programmerat the center.

I understand that Taylor produces both in-house and customer documentation. I am a fresh graduated of DeVry Institute of Technology in Atlanta with Associate's Degree in Computer Science. In addition to havingtalen a broad range of courses, I served as a computer consultant at the college's computer center where I helped train computer users on new system.

I will be happy to meet you at your convenience and discuss how my

education and experience match your needs. You can reach me at my home address, at (919) 233-1552, or at krock@devry.alumni.edu

22. Why did Mr. Krick write the letter?
- To apply for a job.
 - To ask for some information.
 - To introduce himself to Taylor.
 - To explain his educational background.
 - To explain his course.
23. How does Mr. Krick know the vacancy?
- From the radio.
 - From a colleague
 - From the television.
 - From Atlanta.
 - From newspaper
24. What will the Human Resources Director probably do after reading the letter?
- Evaluate the salary offered.
 - Start working at Taylor, Inc.
 - Promote Mr. Krick to be a manager.
 - Finding his information from newspaper.
 - Ask Mr. Krick to come for an interview.
25. What is the synonym of the word 'fresh' ?
- Old.
 - New.
 - Urgent.
 - Present.
 - Soon.

questions 25-28 refer to the following visiting rules

- Regulations for visitors of the Museum Indonesia
- Buy a ticket before entering the gate.
 - Leave all your belongings before entering the building.
 - Eating, Drinking, and smoking are not allowed in the building.
 - Do not turn on the radio or tape recorder inside the building.
 - Touching the antiques is prohibited.
 - Do not take pictures of the antiquities.

26. What does this passage discuss?
- Problems with the antiquities.
 - Rules for the Museum visitors.
 - Services of the Museum.
 - Rules of a building.
 - Rules of a reservation.
27. What should the visitors do before they come into gate?
- Take a picture.
 - Eat and drink.
 - Leave their belongings.
 - Take a handphone.
 - Buy a ticket.
28. The followings are rules for Museum visitors, EXCEPT :
- Don't eat.
 - Don't smoke.
 - Turn off the radio.
 - Touch the antiquities.
 - Don't run
29. What's the antonym of the word 'prohibited' in rule no 5?
- Left.
 - Limited.
 - Forbidden.
 - Exception.
 - Permitted.

questions 29-36 refer to the following notice

eacher. Missing. Mr. Waluyo, age 45, failed to return to his apartment on Saturday night. Mr. Waluyo is 170 cm tall, weights 50 kg and has short black hair. He was wearing a black jacket, jeans and running shoes when last seen. Anyone who saw him please contact Nanny at 64127787 or 0812456099

30. What does the notice tell us about?
- The location of Mr. Waluyo's apartment.
 - An introduction of a new teacher.
 - Nanny, the contact person.
 - A missing person.
 - A finding person
31. What did Mr. Waluyo look like when last seen?
- He was a very young man.

- b. He lived in an apartment.
 - c. He wore sport shoes.
 - d. He had long hair.
 - e. He had black jacket.
32. Which statement is NOT true according to the next?
- a. Mr. Waluyo is teacher.
 - b. Mr. waluyo is overweight.
 - c. Mr. Waluyo left his apartment.
 - d. Mr. Waluyo disappeared on Saturday night.
 - e. Mr. Waluyo goes to apartement.
33. Sheila : What do you do in present company?
Dimas :
- a. I have sent this insurance policy to a new client.
 - b. I check mail, sort out letters, and put them in the agenda.
 - c. I will have to arrange the annual meeting with the stakeholders.
 - d. I met some clients this morning, and go to explain the company policy.
 - e. I would accept a job offer and begin my first job.
34. Dimas : I'm thingking about quitting my job.
Sheila : Sorry? Are you serious? It's so difficult to find a job these days.
Dimas : I know, but.... I'm thingking about continuing my studies.
- a. I'm not going to find another job.
 - b. The job was not interesting anymore.
 - c. I did not want to work for a bigger company.
 - d. The company will try to stop me from walking out.
 - e. I still at home.
35. Waiter : Excuse me. Are you ready to order?
Dimas : Yes, can I have fish and chips, and coke? Sheila?
Sheila :, please.
Waiter : Fish and chips, coke, noodles, and hot tea. I'll be right back.
- a. A glass of coke and a bag of noodles.
 - b. A bag of chips and a plate of noodles.
 - c. A bottle of hot tea and a plate of chips.
 - d. A bottle of hot coffe and a plate of chips.
 - e. A bowl of noodles and a cup of hot tea.
36. Sam : How much is that new Toyota?
Salesperson : It's \$4,700, tax included
Sam :! can't afford it.
- a. The cost is low.
 - b. That's cheap enough.
 - c. That's too expensive.
 - d. The price is quite reasonable.
 - e. That's too cheap for me.

37. Nahya : Raisha, our new staff?
 Budi : Oh, yes. I met her yesterday. Hi, Raisha. How are you doing?
 Raisha : Hi, Budi. I'm doing fine, Thanks. How are you?
- Have you met.
 - Do you know about.
 - Do you want to meet
 - Are you going to meet.
 - Do you visit.
38. Taufiq : Happy Travel, Taufiq speaking.
 Olga : Hello, Mr. Taufiq. This Olga...
 Taufiq : Oh, yes. Miss Olga. I hope you and your family will have a wonderful
 Time.
 Olga : Thanks you. We are very excited.
- I am sad to tell you the truth.
 - I liked to talk about the ticket service.
 - I am calling to confirm my trip to Disneyland.
 - I am agree to tell you the truth.
 - i am disagree to tell you the truth.
39. Mela : I have already put the letter here. Was'nt next?
 Bowo : do you see the "start" button on the machine? If you press it,..
 Mela : Oh, yes. Thank you.
- You will see the next instruction.
 - You would see the next instruction.
 - You would see the next instruction.
 - You would have seen the next instruction.
 - You won't have seen the next instruction
40. Fahri : You've a got big bonus and incentives, haven't you?...? I'm hungry.
 Vita : Okay, where do you usually have lunch
- Why didn't you tell me
 - Why don't you treat me
 - Why didn't you eat lunch
 - Why must we wait for them
 - Why do not you tell me

Lampiran 2. Materi Soal Fisika Aplikasi Simulasi Tes

Soal Simulasi Tes Fisika

1. Bahan dimana suatu arah berfungsi sebagai konduktor dan pada arah yang lain berfungsi sebagai isolator adalah....
A. Konduktor. C. Semikonduktor. E. Semiisolator.
B. Isolator. D. Kondensastor.
2. Gaya tarik menarik dua benda bermuatan listrik besarnya sebanding dengan perkalian masing-masing, muatannya dan berbanding terbalik dengan kuadrat jarak antara kedua muatannya adalah bunyi hukum....
A. Coulomb. C. Amper. E. Kirchoff I.
B. Ohm. D. Faraday
3. Sebuah resistor dengan hambatan $R = 10$ dilewati oleh arus listrik sebesar 0,5, maka beda potensial antara ujung-ujung resistor tersebut adalah..
A. 0,005 volt. C. 0,5 volt E. 5,5 volt
B. 0,05 volt D. 5 volt
4. Jumlah muatan listrik baik muatan positif maupun negatif melewati suatu penampang penghantar persatuan waktu dinamakan...
A. Daya. C. Tegangan E. Kuat arus.
B. Ampere. D. Gaya.
5. Suatu penghantar panjangnya 10 m dengan luas penampang 2 cm dan hambatan jenisnya 10, maka hambatan penghantarnya adalah....
A. 0,05 Ohm. C. 1,0 Ohm E. 2,5 Ohm
B. 0,5 Ohm D. 1,5 Ohm
6. Derajat skala Fahrenheit dan Celcius akan menunjukkan skala yang sama pada...
A. -20 C. -40 E. -60
B. -30 D. -50

7. Sebatang aluminium pada suhu 0°C ukuran panjangnya 1,5 m, suhunya ditambah menjadi 50°C berapa panjangnya sekarang bila koefisien muai panjang $\alpha = 24 \cdot 10^{-6}/^{\circ}\text{C}$?
- A. 1,5012 m C. 1,5120 m E. 1,5281 m
 B. 1,5112 m D. 1,5211 m
8. Sebatang perak dipanaskan sehingga suhunya naik 80°C . Setelah diteliti ternyata perak tersebut bertambah panjang 0.19 mm. Koefisien panjang perak $19 \cdot 10^{-6}/^{\circ}\text{C}$. jika suhu mula-mula perak 30°C berapa panjang mula-mula...
- A. 10 cm C. 20 cm E. 30 cm
 B. 15 cm D. 25 cm
9. Sebuah setrika listrik 350 watt dipakai selama 1 jam. Berapa energi yang terpakai/habis?
- A. 12.600 J
 B. 126.000 J
 C. 1.260.000 J
 D. 12.600.000 J
 E. 126. 000.000 J
10. Pada sebuah lampu pijar tertulis 25 Watt, 220 volt. Berapa hambatan lampu itu?
- A. 1,936 Ohm
 B. 19,36 Ohm
 C. 193,6 Ohm
 D. 1936 Ohm
 E. 2000 Ohm
11. Lampu pijar 100 watt 220 volt. Bila lampu itu dipasang pada tegangan 110 volt, daya lampu tersebut adalah...
- A. 200 Watt
 B. 100 Watt
 C. 75 Watt
 D. 50 Watt
 E. 25 Watt

12. Sebuah transformator memiliki tegangan primer 220 volt dengan jumlah gulungan 100 lilitan, akan dirubah tegangannya menjadi 110 volt. Berapa jumlah lilitan agar sesuai dengan yang diinginkan...
- A. 50 lilitan
 - B. 75 lilitan
 - C. 100 lilitan
 - D. 150 lilitan
 - E. 200 lilitan
13. Tiga buah resistor masing-masing 2 ohm, 4 ohm dan 3 ohm dipasang dalam rangkaian seri, maka tahanan totalnya adalah...
- A. 5 ohm
 - B. 6 ohm
 - C. 7 ohm
 - D. 8 ohm
 - E. 9 ohm
14. Alat yang digunakan untuk menaikkan atau menurunkan tegangan listrik adalah..
- A. Transistor
 - B. Resistor
 - C. Kapasitor
 - D. Kondensator.
 - E. Transformator
15. Bahan dimana banyak elektron dapat dengan mudah melaluinya disebut...
- A. Konduktor
 - B. Semikonduktor.
 - C. Semiisolator.
 - D. Isolator.
 - E. Transistor.
16. Lima buah hambatan masing-masing 10 ohm dipasang paralel, maka hambatan penggantinya adalah..

- A. 3 Ohm
 - B. 4 Ohm
 - C. 5 Ohm
 - D. 6 Ohm
 - E. 7 Ohm
17. Termometer Fahrenheit yang diletakan pada sebuah mesin menunjukan suhu 101°F . Suhu tersebut setara dengan....
- A. $18,3^{\circ}\text{C}$
 - B. $28,3^{\circ}\text{C}$
 - C. $38,3^{\circ}\text{C}$
 - D. $48,3^{\circ}\text{C}$
 - E. $58,3^{\circ}\text{C}$
18. Sebuah pipa air mempunyai luas penampang $2,5\text{ cm}^2$. Jika kecepatan aliran air yang keluar $0,2\text{ m/s}$ maka debit air yang keluar...
- A. $25\text{ cm}^3/\text{s}$
 - B. $50\text{ cm}^3/\text{s}$
 - C. $55\text{ cm}^3/\text{s}$
 - D. $65\text{ cm}^3/\text{s}$
 - E. $70\text{ cm}^3/\text{s}$
19. Sebuah benda dilepaskan tanpa kecepatan awal dari sebuah menara yang tingginya 100 m (gesekan udara diabaikan) jika percepatan gravitasi 10m/dt , maka ketinggian benda diukur dari tanah pada detik ke 2 adalah...
- A. 40 m
 - B. 50 m
 - C. 60 m
 - D. 70 m
 - E. 80 m
20. Suatu fluida bergerak dengan kecepatan 20 cm/s^2 pada pipa dengan diameter 4 cm dilewatkan pada sebuah venturi berdiameter 2 cm berapa kecepatan fluida setelah melewati venturi....
- A. 20 cm/s^2

- B. 30 cm/s^2
 - C. 40 cm/s^2
 - D. 50 cm/s^2
 - E. 60 cm/s^2
21. Berapakah usaha yang diperlukan untuk memindahkan benda yang masanya 5 kg dari lantai ke atas meja yang tingginya 1.2 m bila percepatan gravitasi 10 m/s^2 ...
- A. 30 J
 - B. 40 J
 - C. 50 J
 - D. 60 J
 - E. 70 J
22. Sebuah benda massanya 2 kg mula-mula dalam keadaan diam pada sebuah bidang datar yang licin. Berapakah usaha yang dilakukan pada benda sehingga kecepatannya menjadi 8 m/s
- A. 24 Joule
 - B. 34 Joule
 - C. 54 Joule
 - D. 60 Joule
 - E. 64 Joule
23. Sebuah benda berada pada ketinggian 20 m dari tanah, kemudian benda tersebut jatuh bebas. Berapakah usaha yang dilakukan oleh gaya berat sehingga benda jatuh ke tanah. Jika masa benda 1 kg dan percepatan gravitasi 10 m/s^2 ...
- A. 100 Joule
 - B. 200 Joule
 - C. 300 Joule
 - D. 400 Joule
 - E. 500 Joule

24. Plat besi luasnya 4 m^2 pada suhu 20°C . Bila suhunya dinaikkan menjadi 100°C , berapa luasnya sekarang bila diketahui koefisien muai panjang besi $11 \cdot 10^{-6} / ^\circ \text{C}$...
- A. $4,00716 \text{ m}^2$
 - B. $4,00176 \text{ m}^2$
 - C. $4,00741 \text{ m}^2$
 - D. $4,0704 \text{ m}^2$
 - E. $4,0740 \text{ m}^2$
25. Sebuah peralatan listrik mempunyai hambatan dalam 1 kilo ohm dihubungkan dengan sumber arus 0,5 ampere selama 1 menit, besarnya energi yang dipergunakan adalah...
- A. $1,0 \cdot 10^4 \text{ J}$
 - B. $1,5 \cdot 10^4 \text{ J}$
 - C. $2,0 \cdot 10^4 \text{ J}$
 - D. $2,5 \cdot 10^4 \text{ J}$
 - E. $3,0 \cdot 10^4 \text{ J}$
26. Pada sebuah titik percabangan dari suatu rangkaian listrik, jumlah arus yang masuk sama dengan arus yang keluar adalah...
- A. Hk. Coulomb
 - B. Hk. Ohm
 - C. Hk. Faraday
 - D. Hk. I Kirchoff
 - E. Hk. Amper
27. Sebuah trafo ideal mempunyai tegangan primer dan sekunder 80 volt dan 220 volt, jika arus yang mengalir pada kumparan primer 2 ampere, berapa arus yang mengalir pada kumparan sekunder...
- A. 0,4
 - B. 0,6
 - C. 0,8
 - D. 1,0
 - E. 1,2

28. Ciri-ciri trafo step down adalah..
- A. $R_p < R_s$
 - B. $I_p > I_s$
 - C. $N_p > N_s$
 - D. $V_p < V_s$
 - E. $N_s > N_p$
29. Efisiensi trafo dinyatakan dengan persamaan.....
- A. $\eta = V_s/V_p \times 100 \%$
 - B. $\eta = P_s/P_p \times 100 \%$
 - C. $\eta = V_p/V_s \times 100 \%$
 - D. $\eta = P_p/P_s \times 100 \%$
 - E. $\eta = I_p/I_s \times 100 \%$
30. Besarnya hambatan perngantar tergantung kepada hal-hal berikut ini, kecuali..
- A. Jenis penghantar
 - B. Panjang penghantar
 - C. Gaya tarik
 - D. Suhu
 - E. Tegangan
31. Energy listrik sebesar 746 watt bila dikonversikan ke dalam Horse Power menjadi...
- A. 1 HP
 - B. 2 HP
 - C. 3 HP
 - D. 4 HP
 - E. 5 HP
32. Lima buah elemen dipasang pararel yang masing-masing berharga $E = 1,5$ volt dan $R = 1$ ohm, maka gaya gerak listrik penggantinya dan hambatan penggantinya adalah...
- A. 1,5 volt dan 0,2 ohm
 - B. 1,5 volt dan 1 ohm

- C. 1,5 volt dan 5 ohm
 D. 7,5 volt dan 0,2 ohm
 E. 7,5 volt dan 5 ohm
33. Sebuah benda masanya 1 Kg digantung dengan seutas tali, maka gaya tegangan tali adalah ($g = 9,8 \text{ m/s}^2$)
 A. 0 newton
 B. 4,9 newton
 C. 9,8 newton
 D. 10 newton
 E. 10,8 newton
34. Lampu 100 watt 110 volt bila dipasang pada tegangan 220 akan putus, karena daya lampunya menjadi...
 A. 50 watt
 B. 100 watt
 C. 150 watt
 D. 400 watt
 E. 600 watt
35. Benda bermassa 5 kg yang mula-mula diam dipercepat oleh suatu gaya tetap newton. Setelah menempuh jarak 9 m, kelajuan benda tersebut adalah...
 A. 118 m/s C. 6 m/s E. 3,6 m/s
 B. 36 m/s D. 4.5 m/s
36. Benda bergerak lurus di bawah pengaruh resultan gaya tetap. Selama 4 detik momentum linier benda tersebut berubah dari 4 kg m/s menjadi 12 kg m/s dengan arah gerak akhir berlawanan dengan arah gerak mula-mula. Resultan gaya pada benda itu besarnya...
 A. 2 N C. 8 N E. 12 N
 B. 4 N D. 10 N
37. Massa 2 kg digantung pada pegas yang mempunyai tetapan gaya 1000 N/m, hingga mencapai keadaan diam dan setimbang. Usaha yang diperlukan untuk

mengubah simpangan benda dari posisi setimbangnya dari 2 cm menjadi 8 cm adalah..

- A. 12 J C. 6 J E. 3 J
B. 8 J D. 4 J

38. Sebuah lensa cembung dobel (*double convex*) tipis mempunyai jejari kelengkungan sebesar 40 cm dan dibuat dari kaca dengan $n = 1,65$. Dalam cm panjang fokusnya adalah....

- A. 35 C. 25 E. 20
B. 31 D. 21

39. Potensial listrik sejauh 4 cm dari suatu muatan titik Q sama dengan 10 V. Potensial listrik sejauh R dari muatan titik 5Q sama dengan 20 V. Nilai R sama dengan....

- A. 20 cm C. 12 cm E. 6 cm
B. 15 cm D. 10 cm

40. Sebuah benda 2 kg jatuh bebas dari ketinggian 20 m di atas tanah. Berapa lama waktu yang diperlukam oleh benda untuk mencapai tanah? ($g = 10 \text{ ms}^{-2}$)

- A. 20 s C. 10 s E. 2 s
B. 18 s D. 5 s

Lampiran 3. Materi Soal Fisika Aplikasi Simulasi Tes

Soal Simulasi Tes Kimia

1. Seorang siswa mencampurkan dua macam larutan zat kimia. Di antara pernyataan di bawah ini yang tidak menunjukkan ciri-ciri terjadinya *reaksi kimia* adalah...
A. Timbul gas. C. Perubahan Suhu E. Perubahan warna
B. Terjadinya endapan D. Perubahan Massa
2. Untuk memisahkan alcohol dari air dilakukan melalui proses?
A. Destilasi C. Kristalisasi E. Filtrasi
B. Sublimasi D. Kromatografi
3. Di antara pasangan zat berikut termasuk kelompok senyawa adalah...
A. Tembaga dan belerang. D. Alkohol dan urea
B. Garam dan perunggu E. Phospor dan oksigen
C. Kuningan dan gula
4. Di antara peralatan laboratorium kimia berikut yang hanya diperbolehkan sebagai wadah pemanas cairan adalah...
A. Buret C. Mortar E. Beaker glass
B. Labu erlenmeyer D. Labu takar
5. Jika Ar O = 16, Ar Na = 23, Ar C = 12, dan H = 1, maka massa molekul relative (Mr) dari $\text{Na}_2\text{CO}_3 \cdot 10\text{H}_2\text{O}$ adalah...
A. 106 C. 266 E. 1261
B. 234 D. 286

6. Jika pada STP volume dari 1,6 gram gas sebesar 1,12 liter. Maka massa molekul relatif (Mr) gas tersebut adalah...
- A. 28 C. 32 E. 36
B. 30 D. 34
7. Massa dari $3,01 \times 10^{23}$ atom Ca (Ar = 40) adalah..
- A. 10 gram C. 40 gram E. 80 gram
B. 20 gram D. 60 gram
8. Massa unsur nitrogen yang terkandung dalam 120 gram pupuk urea $[\text{CO}(\text{NH}_2)_2]$ adalah sebanyak... (Ar C = 12, Ar O = 12, Ar N = 14, Ar H = 1)
- A. 28 gram C. 56 gram E. 84 gram
B. 42 gram D. 76 gram
9. Suatu unsur dalam keadaan netral, 1 atomnya mengandung 11 elektron dan 12 neutron, berarti unsur tersebut mempunyai nomor atom dan nomor massa...
- A. 11 dan 12 C. 11 dan 23 E. 23 dan 11
B. 12 dan 11 D. 12 dan 22
10. Jika unsur X memiliki nomor atom, 12 dan nomor massa 24, maka dalam tabel sistem periodik, unsur X terletak pada.....
- A. Golongan II A perioda 3
B. Golongan II A perioda 4
C. Golongan III A perioda 4
D. Golongan IV A perioda 2
E. Golongan IV A perioda 3

11. Dari pasangan-pasangan senyawa di bawah ini, mana yang mempunyai ikatan kovalen pada kedua senyawa....
- A. NH_3 dan KCl C. H_2O dan CCl_4 E. HF dan LiCl
B. CO_2 dan BaCl_2 D. NaCl dan KBr
12. Diketahui unsur-unsur P,Q,R,S dan T dengan nomor atom berturut-turut 12,13,14,15 dan 17 ikatan ion dapat terjadi antara atom-atom unsur...
- A. P dan Q C. R dan S E. S dan T
B. Q dan R D. P dan T
13. Dari pasangan unsur berikut yang merupakan unsur hara makro bagi pertumbuhan tanaman adalah...
- A. Na dan P C. N dan P E. Ca dan C
B. Ni dan C D. Mg dan C
14. Konsentrasi larutan yang mengandung 3 gram pupuk urea [$\text{CO}(\text{NH}_2)_2$] dalam 200 mL larutan adalah... (Ar C = 12, Ar O = 16, Ar N = 14, Ar H = 1)
- A. 0,25 M C. 0,125 M E. 100 M
B. 0,50 M D. 0,375 M
15. Derajat keasaman (pH) asam klorida 0,02 M adalah...
- A. 1 C. 2 E. $2 + \log 2$
B. $1 - \log 2$ D. $2 - \log 2$

16. Laju reaksi $2A + 2B \rightarrow 3C + D$ pada setiap saat dapat dinyatakan sebagai...

- A. Penambahan konsentrasi A tiap satuan waktu.
- B. Penambahan konsentrasi B tiap satuan waktu.
- C. Penambahan konsentrasi C tiap satuan waktu.
- D. Penambahan konsentrasi A dan B tiap satuan waktu.
- E. Penambahan konsentrasi B dan C tiap satuan waktu.

17. Heksana adalah satu senyawa hidrokarbon yang rumus molekulnya adalah....

- A. C_2H_{14} C. C_6H_{10} E. C_7H_{12}
- B. C_6H_{14} D. C_7H_{14}

18. Salah satu contoh aerosol cair yang sering kita jumpai dalam kehidupan sehari-hari adalah...

- A. Kabut C. Mutiara E. Lem
- B. Sabun D. Susu

19. Perhatikan beberapa sifat zat berikut!

- (1) mudah meleleh;
- (2) memiliki inti yang tidak stabil;
- (3) memiliki energy ionisasi yang tinggi; dan
- (4) dapat mengalami peluruhan.

Sifat zat radioaktif terdapat pada nomor....

- A. (1) dan (2) C. (2) dan (3) E. (3) dan (4)
- B. (1) dan (4) D. (2) dan (4)

20. Beberapa contoh penerapan sifat koligatif larutan dalam kehidupan :

- (1) pemberian garam dapur untuk mencairkan salju;
- (2) membuat cairan infuse yang akan digunakan pada tubuh manusia;
- (3) proses penyerapan air oleh akar tanaman dari dalam tanah;
- (4) penggunaan garam dapur pada pembuat es putar; dan
- (5) pembuatan ikan asin.

Penerapan sifat koligatif titik beku larutan adalah nomor

A. (1) dan (2) C. (2) dan (3) E. (4) dan (5)

B. (1) dan (4) D. (3) dan (5)

21. Di antara karbohidrat berikut yang tidak mengalami hidrolisis adalah....

A. Fruktosa C. Sukrosa E. Selulosa

B. Maltose D. amilum

22. Kelompok senyawa berikut yang terdiri dari monosakarida, disakarida, dan polisakarida berurutan adalah.....

A. Laktosa, fruktosa, selulosa

B. Glukosa, maltose, amilum

C. Glikogen, maltose, selulosa

D. Amilum, selulosa, glikogen

E. Galaktosa, maltose, sukrosa

23. Semua jenis karbohidrat dalam tubuh dengan bantuan enzim akan diuraikan menjadi senyawa....

A. Laktosa C. Sukrosa E. Fruktosa

B. Glukosa D. Maltosa

24. Pencemaran gas karbon dioksida menimbulkan kerugian terhadap manusia yaitu...

- A. Menimbulkan hujan asam
- B. Tergangunya penyerapan O_2 oleh Haemoglobin dalam darah.
- C. Merusak lapisan ozon.
- D. Menimbulkan kerusakan pada kulit.
- E. Menyebabkan suhu udara naik.
25. Hasil analisis air di daerah X sukar menghasilkan buih bila digunakan untuk mencuci. Diperkirakan di daerah tersebut mengandung ion...
- A. Na^+ , K^+ C. Ca^{2+} ; Mg^{2+} E. K^+ , Ca^{2+}
- B. Fe^{2+} , Mg^{2+} D. Cu^{2+} ; Ha^+
26. Air sadah yang dapat dilunakan dengan cara pemanasan adalah air mengandung garam....
- A. $Ca(HCO_3)_3$ C. $CaCl_2$ E. $CaSO_4$
- B. $MgSO_4$ D. $MgCl_2$
27. Isotop radioaktif yang digunakan dalam reaksi fotosintesis adalah....
- A. Co-60 C. C-14 E. C1-37
- B. N-15 D. Na-24
28. Perhatikan perubahan materi yang terjadi di bawah ini :
- (1) Dinamit diledakan
- (2) Pembakaran bensin
- (3) Penyepuhan logam
- (4) Pemurnian air dengan distilasi
- (5) Air membeku.
- Perubahan materi yang merupakan perubahan kimia adalah...
- A. 1,3,5 C. 2, 3, 5 E. 3, 4, 5
- B. 1, 2, 3 D. 2, 4, 5
29. Lambang unsur seng, tembaga dan timbale berturut- turut adalah...

- A. Ti, Te, Cu C. Zn,Cu, Pb E. Pt, Pb, Sn
- B. Sn, Te, Ti D. Pb, Cu, Sn
30. Untuk memisahkan gula tebu dari air tebu dilakukan melalui proses...
- A. Filtrasi C. Kristalisasi E. Distilasi
- B. Sublimasi D. Kromatografi
31. Di antara pasangan zat berikut termasuk yang termasuk kelompok unsur...
- A. tembaga dan belerang C. kuningan dan gula E. alcohol dan urea
- B. garam dan perunggu D. phosphor dan gas oksigen
32. Di antara peralatan laboratorium kimia berikut yang digunakan membuat larutan secara kuantitatif adalah...
- A. Buret C. Beaker glass E. Mortar
- B. Labu Erlenmeyer D. Labu Takar
33. Dalam molekul $2\text{Al}(\text{OH})_3$ terdapat.....
- A. 4 atom Al, 6 atom O, 6 atom H
- B. 4 atom Al, 3 atom O, 3 atom H
- C. 6 atom Al, 3 atom O, 3 atom H
- D. 2 atom Al, 3 atom O, 3 atom H
- E. 2 atom Al, 1 atom O, 3 atom H
34. Pada reaksi :
- $$a \text{N}_2 + b \text{H}_2 \rightarrow c \text{NH}_3$$
- Koefisien a, b, dan c secara berurutan pada persamaan reaksi di atas....
- A. 1,2,3 C. 2,3,1 E. 3,2,1
- B. 1,3,2 D. 3,1,2

35. Perhatikan kegunaan bahan makan dalam tubuh!

- (1) Biokatalis.
- (2) Sumber energy primer.
- (3) Zat pembangun.
- (4) Pelarut vitamin.

Fungsi dari protein ditunjukkan pada nomor...

- A. (1) dan (4) C. (2) dan (3) E. (3) dan (4)
- B. (1) dan (3) D. (2) dan (4)

36. Beberapa kegunaan senyawa karbon sebagai:

- (1) pemberi aroma;
- (2) pengawet preparat;
- (3) bahan bakar;
- (4) pelarut; dan
- (5) pembersih cat kuku.

Kegunaan dari aseton terdapat pada nomor....

- A. (1) dan (2) C. (2) dan (3) E. (4) dan (5)
- B. (1) dan (3) D. (3) dan (4)

37. Berikut ini adalah penerapan sifat koloid dalam kehidupan sehari-hari.

- (1) Pemutihan gula dengan arang tulang.
- (2) Langit berwarna jingga pada petang hari.
- (3) Penyerapan asap pabrik dengan alat Cottrel.
- (4) Proses cuci darah pada penderita gagal ginjal.
- (5) Pengumpulan karet oleh lateks.

Contoh penerapan dalam sehari-hari sifat adsorpsi terdapat pada nomor..

- A. (1) dan (3) C. (2) dan (4) E. (4) dan (5)
- B. (2) dan (3) D. (3) dan (4)

38. Berikut ini beberapa senyawa turunan benzena yang dapat dimanfaatkan dalam kehidupan sehari-hari.

- (1) asam benzoat;
- (2) anilina;
- (3) fenol; dan

(4) stirena.

- A. (1) dan (3) C. (2) dan (3) E. (3) dan (4)
B. (1) dan (4) D. (2) dan (4)

39. Dari pasangan-pasangan senyawa di bawah ini, mana yang mempunyai ikatan ion pada kedua senyawa....


- A. NH_3 dan CCl_4 C. H_2O dan CCl_4 E. H_2 dan LiCl
B. CO_2 dan Cl_2 D. NaCl dan KBr

40. Larutan $\text{Ca}(\text{OH})_2$ 0,01 M sebanyak 1 L memiliki pH sebesar....

- A. $2 - \log 2$ C. $12 - \log 2$ E. $12 + \log 2$
B. $2 + \log 1$ D. $12 + \log 1$


Lampiran 4. Surat Ijin Penelitian Fakultas

07/02/2012 13:24:00



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI YOGYAKARTA
FAKULTAS TEKNIK

Alamat : Kampus Karangmalang, Yogyakarta, 55281
Telp. (0274) 586168 ps.w. 276,289,292 (0274) 586734 Fax. (0274) 586734
website : <http://ft.uny.ac.id> e-mail: ft@uny.ac.id ; teknik@uny.ac.id



Certificate No. QSC 00582

Nomor : 0172/UN34.15/PL/2012
Lamp. : 1 (satu) bendel
Hal : Permohonan Ijin Penelitian

07 Februari 2012

Yth.

1. Gubernur Provinsi DIY c.q. Ka. Biro Administrasi Pembangunan Setda Provinsi DIY
2. Walikota Yogyakarta c.q. Kepala Dinas Perijinan Kota Yogyakarta
3. Kepala Dinas Pendidikan, Pemuda dan Olahraga Propinsi DIY
4. Kepala Dinas Pendidikan Kota Yogyakarta
5. KEPALA SMK N 3 YOGYAKARTA

Dalam rangka pelaksanaan Tugas Akhir Skripsi kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan penelitian dengan judul **"APLIKASI SIMULASI TES BERBASIS MOBILE APPLICATION ENGGUNAKAN BAHASA PEMROGRMAN JAVA MICRO EDITION, PHP DAN MYSQL"**, bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta tersebut di bawah ini:

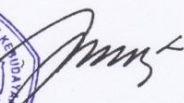
No.	Nama	NIM	Jurusan/Prodi	Lokasi Penelitian
1	Rizam Yudinar	08502244021	Pend. Teknik Elektronika - S1	SMK N 3 YOGYAKARTA

Dosen Pembimbing/Dosen Pengampu : Totok Sukardiyono, MT.
NIP : 19670930 199303 1 005


Adapun pelaksanaan penelitian dilakukan mulai tanggal 07 Februari 2012 sampai dengan selesai.

Demikian permohonan ini, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.

Dekan,
Wakil Dekan I,



Dr. Sunaryo Soenarto
NIP. 19580630 198601 1 001



Tembusan:
Ketua Jurusan

08502244021 No. 912

Lampiran 5. Surat Ijin Penelitian BAPPEDA



**PEMERINTAH PROVINSI DAERAH ISTIMEWA YOGYAKARTA
SEKRETARIAT DAERAH**

Kompleks Kepatihan, Danurejan, Telepon (0274) 562811 - 562814 (Hunting)
YOGYAKARTA 55213

SURAT KETERANGAN / IJIN

070/7031/N/7/2012

Membaca Surat : Dekan Fak. Teknik UNY

Nomor : 0172/UN.34.15/PL/2012

Tanggal : 07 Februari 2012

Perihal : Ijin Penelitian

Mengingat : 1. Peraturan Pemerintah Nomor 41 Tahun 2006, tentang Perizinan bagi Perguruan Tinggi Asing, Lembaga Penelitian dan Pengembangan Asing, Badan Usaha Asing dan Orang Asing dalam melakukan Kegiatan Penelitian dan Pengembangan di Indonesia;
2. Peraturan Menteri Dalam Negeri Nomor 33 Tahun 2007, tentang Pedoman penyelenggaraan Penelitian dan Pengembangan di Lingkungan Departemen Dalam Negeri dan Pemerintah Daerah;
3. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 37 Tahun 2008, tentang Rincian Tugas dan Fungsi Satuan Organisasi di Lingkungan Sekretariat Daerah dan Sekretariat Dewan Perwakilan Rakyat Daerah.
4. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 18 Tahun 2009 tentang Pedoman Pelayanan Perizinan, Rekomendasi Pelaksanaan Survei, Penelitian, Pendataan, Pengembangan, Pengkajian, dan Studi Lapangan di Daerah Istimewa Yogyakarta.

DIIJINKAN untuk melakukan kegiatan survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan kepada:

Nama : RIZAM YUDINAR

NIP/NIM : 08502244021

Alamat : Karangmalang Yogyakarta

Judul : APLIKASI SIMULASI TES BERBASIS MOBILE APPLICATION MENGGUNAKAN BAHASA PEMROGRAMAN JAVA MICRO EDITION, PHP DAN MYSQL

Lokasi : SMK N 3 YOGYAKARTA Kota/Kab. KOTA YOGYAKARTA

Waktu : 31 Juli 2012 s/d 31 Oktober 2012

Dengan Ketentuan

1. Menyerahkan surat keterangan/ijin survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan *) dari Pemerintah Provinsi DIY kepada Bupati/Walikota melalui institusi yang berwenang mengeluarkan ijin dimaksud;
2. Menyerahkan soft copy hasil penelitiannya baik kepada Gubernur Daerah Istimewa Yogyakarta melalui Biro Administrasi Pembangunan Setda Provinsi DIY dalam compact disk (CD) maupun mengunggah (upload) melalui website adbang.jogjaprov.go.id dan menunjukkan cetakan asli yang sudah disahkan dan dibubuhi cap institusi;
3. Ijin ini hanya dipergunakan untuk keperluan ilmiah, dan pemegang ijin wajib mentaati ketentuan yang berlaku di lokasi kegiatan;
4. Ijin penelitian dapat diperpanjang maksimal 2 (dua) kali dengan menunjukkan surat ini kembali sebelum berakhir waktunya setelah mengajukan perpanjangan melalui website adbang.jogjaprov.go.id;
5. Ijin yang diberikan dapat dibatalkan sewaktu-waktu apabila pemegang ijin ini tidak memenuhi ketentuan yang berlaku.

Dikeluarkan di Yogyakarta

Pada tanggal 31 Juli 2012

A.n Sekretaris Daerah

Asisten Perekonomian dan Pembangunan
Ub.

Kepala Biro Administrasi Pembangunan



Tembusan :

1. Yth. Gubernur Daerah Istimewa Yogyakarta (sebagai laporan);
2. Walikota Yogyakarta c.q Ka. Dinas Perizinan
3. Ka. Dinas Pendidikan, Pemuda & OR Prov. DIY
4. Dekan Fak. Teknik UNY
5. Yang bersangkutan

Lampiran 6. Surat Pernyataan Validasi

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Herman Dwi Surjono, Drs., M.Sc., MT., Ph.D.

Jabatan : Lektor Kepala

Telah membaca instrumen penelitian yang berjudul "Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL", oleh:

Nama : Rizam Yudinar

NIM : 08502244021

Prodi : Pendidikan Teknik Elektronika UNY

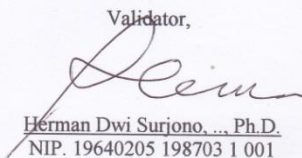
Setelah memperhatikan kisi-kisi instrumen, variabel, indikator, dan butir instrumen maka masukan untuk penulis adalah:

- aspek correctness untuk konten perlu perbaikan

Demikian keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Yogyakarta, Maret 2014

Validator,


Herman Dwi Surjono, ..., Ph.D.
NIP. 19640205 198703 1 001

Lampiran 7. Surat Pernyataan Validasi

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Rahmatul Irfan, S.T, M.T

Jabatan : Asisten Ahli

Telah membaca instrumen penelitian yang berjudul "Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL", oleh:

Nama : Rizam Yudinar

NIM : 08502244021

Prodi : Pendidikan Teknik Elektronika UNY

Setelah memperhatikan kisi-kisi instrumen, variabel, indikator, dan butir instrumen maka masukan untuk penulis adalah:

- ter lampir

.....

.....

.....


.....

.....

Demikian keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Yogyakarta, 12 Mei 2014

Validator,



Rahmatul Irfan, S.T, M.T
NIP.19790517 200604 1 002

Lampiran 8. Pernyataan Validasi Materi Kimia

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Dra. Rr. Sri Handayani
Jabatan : Guru Madya

Telah membaca instrumen penelitian yang berjudul "Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL", oleh:

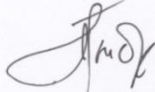
Nama : Rizam Yudinar
NIM : 08502244021
Prodi : Pendidikan Teknik Elektronika UNY

Setelah memperhatikan kisi-kisi instrumen, variabel, indikator, dan butir instrumen maka masukan untuk penulis adalah:

1) KEDAPANNYA BUTA MEMAKAI SISTEM
OPERASI T5 LAIN

.....
.....
.....
.....
.....
.....

Demikian keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Yogyakarta,September 2014
Validator,

Dra. Rr. Sri Handayani
NIP.19630731 198903 2 007

Lampiran 9. Pernyataan Validasi Materi Bahasa Inggris

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Dra. Veronika Rita Rini Wijaya

Jabatan : Guru Pembina

Telah membaca instrumen penelitian yang berjudul "Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL", oleh:

Nama : Rizam Yudinar

NIM : 08502244021

Prodi : Pendidikan Teknik Elektronika UNY

Setelah memperhatikan kisi-kisi instrumen, variabel, indikator, dan butir instrumen maka masukan untuk penulis adalah:

Adanya kesesuaian isi materi soal dengan
syllabus Kurikulum terbaru (mohon disesuaikan)

Demikian keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Yogyakarta,September 2014

Validator,



Dra. Veronika Rita Rini Wijaya
NIP.19621201 199003 2 004

Lampiran 10. Pernyataan Validasi Materi Fisika

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Agus Manaji, S.Si

Jabatan : Guru Muda

Telah membaca instrumen penelitian yang berjudul "Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL", oleh:

Nama : Rizam Yudinar

NIM : 08502244021

Prodi : Pendidikan Teknik Elektronika UNY

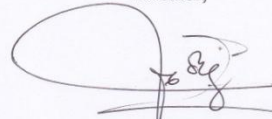
Setelah memperhatikan kisi-kisi instrumen, variabel, indikator, dan butir instrumen maka masukan untuk penulis adalah:

*Masalah Soal Fisika sudah bagus, memenuhi syarat
bagi instrumen penelitian.*

Demikian keterangan ini dibuat agar dapat digunakan sebagaimana mestinya.

Yogyakarta, 24 Juni 2014

Validator,



Agus Manaji, S.Si

NIP.19790316 200604 1 013

Lampiran 11. Revisi Kisi-Kisi dan Instrumen *Reliability*

Kisi-Kisi Instrumen Penelitian Reliability

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Reliability (Reliabilitas)	Consistency (konsistensi)	The use of uniform design and documentation techniques throughout the software development protocol	Penggunaan satu bentuk yang seragam pada seluruh rancangan sistem.	Product Operation
	Accuracy (ketepatan)	The capability of the software product to provide the right or agreed results or effects with the needed degree of precision. Also, the precision of computations and control". the attributes of software that bear on the provision of right or agreed results or effects"	Menunjukkan kemampuan software untuk memberikan hak kontrol atau hasil dan efek (perhitungan) dengan tingkat presisi yang standar atau yang telah ditetapkan.	
	Error tolerance (Toleransi kesalahan)	The damage that occurs when the program encounters an error"	Menunjukkan indikasi kesalahan proses ketika program sedang berjalan.	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.	
	Simplicity (Kesederhanaan)	"The degree to which a program can be understood without difficulty"	Menunjukkan tingkat sejauh mana <i>user</i> dapat memahami program tanpa mengalami kesulitan.	

Menu :

logika
pilihan soal

Tabel. Instrument Aspek Reliability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Accuracy	Apakah <i>splash screen</i> dapat berjalan sesuai resolusi layar? ✓		
2.	Accuracy	Apakah <i>username</i> dapat digunakan untuk <i>login</i> aplikasi? ✓		
3.	Accuracy	Apakah <i>password</i> dapat digunakan untuk <i>login</i> aplikasi? ✓		
4.	Accuracy	Apakah menu pilihan soal dapat ditentukan sesuai keinginan <i>user</i> ? ✓		
5.	Accuracy	Apakah pilihan jawaban dapat ditentukan oleh <i>user</i> menggunakan <i>radiobutton</i> ? ✓		
6.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?		
7.	Error Tolerance	Apakah muncul <i>pop-up</i> pesan kesalahan apabila terjadi kesalahan pengisian data? ✓		
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ? ✓		
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda? ✓		
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ? ✓		
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan? ✓		

Apakah halaman login pada aplikasi berjalan sesuai rancangan?

Apakah menu/halaman login mudah dijalankan?
— 1 — sistem sed m

Lampiran 12. Revisi Kisi-Kisi dan Instrumen *Efficiency*

Kisi-Kisi Instrumen Penelitian Efficiency

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Efficiency (Efisiensi)	Conciseness (Keringkasan)	The compactness of the program in terms of lines of code". Also, Attributes of software that provide the implementation of a function with minimum amount of code".	Menunjukkan bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang relatif ringkas.	Product Operation
	Execution Efficiency (Efektivitas Eksekusi)	The run-time performance of the program	Menunjukkan bahwa eksekusi program berjalan secara efektif.	
	Operability (Operabilitas)	The nature of the software associated with the required amount of effort the user to operate and control software	Menunjukkan bahwa pengguna membutuhkan banyak langkah untuk mengontrol dan mengoperasikan sesuai karakter program yang dikembangkan.	

Tabel. Instrument Aspek *Efficiency*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisennes	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?		
2.	Concisennes	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?		
3.	Concisennes	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?		
4.	Execution Efficiency	Apakah performa aplikasi ^{ekstern} memerlukan banyak waktu?		
5.	Execution Efficiency	Apakah proses pemanggilan data pada aplikasi memerlukan <u>banyak waktu</u> ?		
6.	Operability	Apakah aktivitas program terlalu komplek?		
7.	Operability	Apakah alur penggunaan aplikasi terlalu rumit?		
8.	Operability	Apakah navigasi pada aplikasi terlalu komplex?		

Lampiran 13. Revisi Kisi-Kisi dan Instrumen *Integrity*

Kisi-Kisi Instrumen Penelitian Integrity

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Integrity (Integritas)	Auditability (Audiabilitas)	"The ease with which conformance to standards can be checked"	Menunjukkan kemudahan program program untuk diuji sesuai standar.	Product Operation
	Instrumentation (Instrumentasi)	"The degree to which the program monitors its own operation and identifies errors that do occur"	Sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi	
	Security (Keamanan)	"The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them"	Kemampuan program untuk melindungi informasi dan data sehingga orang atau system yang tidak sesuai tidak dapat membaca atau memodifikasi program dan hanya orang atau system berwenang	

Consistency

1. 1

Tabel. Instrument Aspek *Integrity*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik? <i>berfungsi?</i>		
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik? <i>berfungsi?</i>		
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?		
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?		
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?		
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?		
7.	Instrumentation	Apakah aplikasi dikembangkan berdasarkan penerapan keilmuan?		
8.	Security	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?		
9.	Security	Apakah " <i>admin</i> " dapat menambah <i>dot</i> " <i>Guru</i> " dan " <i>Siswa</i> " baru ?		
10.	Security	Apakah " <i>admin</i> " dapat melakukan perubahan data pada " <i>Guru</i> " dan " <i>Siswa</i> " ? <i>rumah-r.</i>		
11.	Security ✓	Apakah " <i>admin</i> " dapat melihat nilai " <i>Guru</i> " dan " <i>Siswa</i> "? ✓		
12.	Security ✓	Apakah " <i>Guru</i> " dapat melihat nilai " <i>Siswa</i> " ? <i>manajemen</i>		

Apakah user yg memiliki hak akses sebagai admin dapat menambah data Gur- dan Siswa baru ?

No	Kategori	Pertanyaan	Ya	Tidak
13.	Security	Apakah "Siswa" dapat melihat nilai siswa ^{nya} ?		
14.	Security	Apakah "Guru" (tidak login) tidak dapat masuk aplikasi dan mengakses halaman web? Pada ^X		
15.	Security	Apakah "Siswa" (tidak login) tidak dapat masuk aplikasi dan mengakses halaman web? ^X		
16.	Security	Apakah "Guru" (tidak login) tidak dapat melihat data ^{nya} ?		
17.	Security	Apakah "Siswa" (tidak login) tidak dapat melakukan perubahan data.		
18.	Security	Apakah "Siswa" (tidak login) tidak dapat melakukan pengayaan nilai?		

Apakah Guru (tidak login) tidak dapat ~~masuk aplikasi dan~~ bisa mengakses halaman yang mempunyai nilai tersebut?

17. Hal dapat melihat siswa?

Lampiran 14. Revisi Kisi-Kisi dan Instrumen *Maintanability*

Kisi-Kisi Instrumen Penelitian Mainatanability

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Maintanability (Maintanabilitas)	Concisenness (Keringkasan)	"The compactness of the program in terms of lines of code". Also, Attributes of software that provide the implementation of a function with minimum amount of code".	Menunjukkan bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang minimum.	Product Revision
	Consistency (konsistensi)	The use of uniform design and documentation techniques throughout the software development protocol	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	
	Instrumentation (Instrumentasi)	"The degree to which the program monitors its own operation and identifies errors that do occur"	Sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi.	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.	
	Self-Documentation (Dokumentasi diri)	"The degree to which the source code provides meaningful documentation"	Sejauh mana kode sumber menyediakan dokumentasi yang berarti	
	Simplicity (Kesederhanaan)	"The degree to which a program can be understood without difficulty"	Menunjukkan tingkat sejauh mana <i>user</i> dapat memahami program tanpa mengalami kesulitan.	

Tabel. Instrument Aspek Maintainability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Conciseness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana? <i>optimal</i>		
2.	Conciseness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi? <i>optimal</i> ✓		
3.	Conciseness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi? ✓		
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?		
5.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program? ✓		
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ? ✓		
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda? ✓		
8. <i>9</i>	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur? ✓		
9. <i>8</i>	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?		

No	Kategori	Pertanyaan	Ya	Tidak
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?		
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?		
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?		
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?		
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		

Lampiran 15. Revisi Kisi-Kisi dan Instrumen *Flexibility*

Kisi-Kisi Instrumen Penelitian Flexibility

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Flexibilitas (Flexibilitas)	Concisenness (Keringkasan)	"The compactness of the program in terms of lines of code". Also, Attributes of software that provide the implementation of a function with minimum amount of code".	Menunjukkan bahwa implementasi program maupun atribut sebagai fungsi pada aplikasi, dikembangkan dengan jumlah baris kode yang minimum	Product Revision
	Consistency (konsistensi)	The use of uniform design and documentation techniques throughout the software development protocol	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	
	Expandability (Upgradabilitas)	The degree to which architectural, data or procedural design can be extended	Menunjukkan sejauh mana tingkat standar arsitektur, data atau desain prosedural dapat dikembangkan.	
	Generality (Generalitas)	"The breadth of potential application of program components". Generality is defined as the degree to which a software product can perform a wide range of functions.	Menunjukkan tingkat standar generalitas program ditinjau dari sumber (baris kode, attribute, dll) dapat dikembangkan untuk ke tahap selanjutnya.	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.	
	Self-Documentation (Dokumentasi diri)	"The degree to which the source code provides meaningful documentation"	Menunjukkan sejauh mana kode sumber program menyediakan dokumentasi yang berarti.	
	Simplicity (Kesederhanaan)	"The degree to which a program can be understood without difficulty"	Menunjukkan tingkat sejauh mana <i>user</i> dapat memahami program tanpa mengalami kesulitan.	

Tabel. Instrument Aspek *Flexibility*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Conciseness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana? ✓		
2.	Conciseness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi? ✓		
3.	Conciseness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi? ✓		
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?		
5.	Expandability	Dapatkah aplikasi dikembangkan dengan data yang lebih kompleks? ✓		
6.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi? ✓		
7.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?		
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ? ✓		
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda? ✓		
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur? ✓		

5

No	Kategori	Pertanyaan	Ya	Tidak
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?		
12.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?		
13.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?		
14.	Simplicity	Apakah tampilan sistem sangat memudahkan?		
15.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?		
16.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		

Lampiran 16. Revisi Kisi-Kisi dan Instrumen *Testability*

Kisi-Kisi Instrumen Penelitian Testability

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Testability (Testabilitas)	Auditability (Audiabilitas)	"The ease with which conformance to standards can be checked"	Menunjukkan kemudahan program program untuk diuji sesuai standar.	Product Revision
	Instrumentation (Instrumentasi)	"The degree to which the program monitors its own operation and identifies errors that do occur"	Sejauh mana program dapat memonitor sendiri operasi dan mengidentifikasi kesalahan yang memang terjadi ✓	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.	
	Self-Descriptiveness (Deskripsi diri)	"Code possesses the characteristic self-descriptiveness to the extent that it contains enough information for a reader to determine or verify its objectives, assumptions, constraints, inputs, outputs, components, and revision status."	Menunjukkan sejauh mana program mengandung informasi yang cukup bagi <i>user</i> untuk menentukan atau memverifikasi tujuan, asumsi, masukan, komponen, dll.	
	Simplicity (Kesederhanaan)	"The degree to which a program can be understood without difficulty"	Menunjukkan tingkat sejauh mana <i>user</i> dapat memahami program tanpa mengalami kesulitan.	

Pak Irfan

081227210230

Tabel. Instrument Aspek Testability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?		
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?		
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?		
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?		
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?		
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program? ✕	→	revisi folder
7.	Self-Descriptivness	Apakah konten pada program menunjukkan tujuan pembuatan aplikasi? ✕		
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ? ✓		
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda? ✓		
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?		
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?		
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?		

Apakah aplikasi dapat memberikan data informasi program tipe data yang bisa diterima ?

No	Kategori	Pertanyaan	Ya	Tidak
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?		
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		

d) Hardware Independence.

1.

2. Apakah aplikasi simulasi tes dapat berjalan dengan baik pada konfigurasi hardware yang berbeda?

Software System Independence

1.

Data Commonality.

1. Apakah penggunaan tipe data dalam aplikasi sudah sesuai kebutuhan? ✓

Lampiran 17. Revisi Kisi-Kisi dan Instrumen *Reusability*

Kisi-Kisi Instrumen Penelitian Reusability

Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Reusability (Reusabilitas)	Generality (Generalitas)	"The breadth of potential application of program components". Generality is defined as the degree to which a software product can perform a wide range of functions.	Menunjukkan tingkat standar generalitas program ditinjau dari sumber (baris kode, attribute, dll) dapat dikembangkan untuk ke tahap selanjutnya. ✕	Product Transition
	Hardware Independence (Hardware Terpisah)	The degree to which the software is decoupled from the hardware on which it operates	Menunjukkan tingkat kemampuan program atau perangkat lunak untuk dipisahkan dari hardware.	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi.	
	Self-Documentation (Dokumentasi diri)	"The degree to which the source code provides meaningful documentation"	Menunjukkan sejauh mana kode sumber program menyediakan dokumentasi yang berarti.	
	Software system independence (Sistem Software Tersendiri)	The degree to which the program is independent of non-standard programming language features, operating system characteristics, and other environmental constraints	Menunjukkan tingkat dimana program ini bersifat independen (jauh dari bahasa pemrograman yang tidak standar) dan tidak mengalami kendala pada lingkungan sistem operasi.	

ST → 2012
↳ Android

1. Windows → OS
2. Android v → OS
3. ↳ JME

1. tingkat dimana aplikasi tidak tergantung pada fitur dari bahasa pemrograman yang standar, karakteristik OS, dan kendala lingkungan?
1. apakah aplikasi dapat berjalan pada OS yang berbeda (OS interoperability)?
2. apakah aplikasi dapat dikembangkan menggunakan lebih dari satu bahasa pemrograman?

Tabel. Instrument Aspek *Reusability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi? ✗		
2.	Hardware Independence	Apakah aplikasi Simulasi Tes dapat diinstal? ✓		
3.	Hardware Independence ✗	Apakah <i>splash screen</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?		
4.	Hardware Independence ✗	Apakah halaman <i>login</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?		
5.	Hardware Independence ✗	Apakah menu pilihan soal dapat diakses pada <i>device</i> yang berbeda konfigurasi?		
6.	Hardware Independence ✗	Apakah halaman pilihan jawaban dapat diakses pada <i>device</i> yang berbeda konfigurasi?		
7.	Hardware Independence ✗	Apakah <i>upload</i> nilai dapat diakses pada <i>device</i> yang berbeda konfigurasi?		
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ? ✓		
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda? ✓		

1. ~~Apakah aplikasi dapat mesin hard~~
 Apakah fungsi-fungsi yg ada di aplikasi mesin handel
 ketika diimplementasikan ke lingkungan yg lebih luas?

No	Kategori	Pertanyaan	Ya	Tidak
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur? ✓		
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi? ✓		
12.	Software System Independence	Apakah aplikasi dikembangkan sesuai dengan bahasa pemrograman yang mendukung? ✗		

Lampiran 18. Revisi Kisi-Kisi dan Instrumen *Interoperability*

Kisi-Kisi Instrumen Penelitian Interoperability




Software Quality Mc'Call

Software Quality Metric	Quality Factor (sub-characteristic)	Explanation	Translate	Quality Factor
Interoperability (Interoperabilitas)	Communication commonality (Kesamaan Komunikasi) ✓	The degree to which standard interfaces, protocols and bandwidths are used	Menunjukkan bahwa tingkat standar antarmuka, <i>protocol</i> (domain), kapasitas <i>bandwidth</i> yang digunakan untuk pengembangan aplikasi	Product Transition
	Data Commonality (Kesamaan Data)	The use of standard data structures and types throughout the program	Menunjukkan tingkat penggunaan struktur data maupun jenis data yang digunakan telah memenuhi standar pengembangan program.	
	Generality (Generalitas)	"The breadth of potential application of program components". Generality is defined as the degree to which a software product can perform a wide range of functions. ✓	Menunjukkan tingkat standar generalitas program ditinjau dari sumber (baris kode, attribute, dll) dapat dikembangkan untuk ke tahap selanjutnya. ✓	
	Modularity (Modularitas)	"The functional independence of program components". Modularity is increased when it is possible to divide each components into sub-components	Menunjukkan tingkat standar program ditinjau dari modularitas program yang dikembangkan ke dalam masing-masing komponen atau fungsi. ✓	

1. Communication Commonality :

- Antarmuka
 - warna → kelengkapan.
 - Apakah komposisi warna yang ada di aplikasi ? ✓ ?
 - bentuk menu
 - Apakah bentuk menu konsisten ? ✓
 - umpan balik
 - Apakah aplikasi memberikan umpan balik kepada pengguna ketika ~~akan melakukan~~ user melakukan action ?
- protocol
 - Apakah aplikasi mempunyai alur pengembangan ?
- bandwidth
 - Apakah aplikasi dapat berjalan dengan bandwidth yang efisien ?

Tabel. Instrument Aspek *Interoperability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Communication commonality	Apakah tampilan antarmuka aplikasi sesuai untuk standar pengembangan perangkat lunak? 		
2.	Communication commonality	Apakah aplikasi menggunakan protocol sebagai pengembangan? 		
3.	Communication commonality	Apakah aplikasi memerlukan <i>bandwith</i> untuk pengembangan? 		
4.	Data Commonality (Kesamaan Data)	Apakah komponen <i>database</i> dalam aplikasi sudah sesuai dalam penggunaannya?		
5.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?		
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?		
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		

Lampiran 19. Hasil Uji *Reliability*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesanto Jaeni Hantoro
 Pekerjaan : Programmer
 Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Reliability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Accuracy	Apakah <i>splash screen</i> dapat berjalan sesuai resolusi layar?	✓	
2.	Accuracy	Apakah <i>username</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
3.	Accuracy	Apakah <i>password</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
4.	Accuracy	Apakah menu pilihan soal dapat ditentukan sesuai keinginan <i>user</i> ?	✓	
5.	Accuracy	Apakah pilihan jawaban dapat ditentukan oleh <i>user</i> menggunakan <i>radiobutton</i> ?	✓	
6.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Error Tolerance	Apakah muncul pop-up pesan kesalahan apabila terjadi kesalahan pengisian data?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Aryandanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nokia E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Reliability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Accuracy	Apakah <i>splash screen</i> dapat berjalan sesuai resolusi layar?	✓	
2.	Accuracy	Apakah <i>username</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
3.	Accuracy	Apakah <i>password</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
4.	Accuracy	Apakah menu pilihan soal dapat ditentukan sesuai keinginan <i>user</i> ?	✓	
5.	Accuracy	Apakah pilihan jawaban dapat ditentukan oleh <i>user</i> menggunakan <i>radiobutton</i> ?	✓	
6.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Error Tolerance	Apakah muncul <i>pop-up</i> pesan kesalahan apabila terjadi kesalahan pengisian data?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?		✓
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO RIZKY J
Pekerjaan : IT CONSULTANT
Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek Reliability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Accuracy	Apakah <i>splash screen</i> dapat berjalan sesuai resolusi layar?	✓	
2.	Accuracy	Apakah <i>username</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
3.	Accuracy	Apakah <i>password</i> dapat digunakan untuk <i>login</i> aplikasi?	✓	
4.	Accuracy	Apakah menu pilihan soal dapat ditentukan sesuai keinginan <i>user</i> ?	✓	
5.	Accuracy	Apakah pilihan jawaban dapat ditentukan oleh user menggunakan <i>radiobutton</i> ?	✓	
6.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Error Tolerance	Apakah muncul <i>pop-up</i> pesan kesalahan apabila terjadi kesalahan pengisian data?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

Lampiran 20. Hasil Uji *Efficiency*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesapto Joeni Hanfordo
 Pekerjaan : Programmer
 Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Efficiency*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisennes	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisennes	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisennes	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Execution Efficiency	Apakah performa aplikasi memerlukan banyak waktu?	✓	
5.	Execution Efficiency	Apakah proses pemanggilan data pada aplikasi memerlukan banyak waktu?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Operability	Apakah aktivitas program terlalu komplek?	✓	
7.	Operability	Apakah alur penggunaan aplikasi terlalu rumit?	✓	
8.	Operability	Apakah navigasi pada aplikasi terlalu komplek?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Argyudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nela E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*.

Tabel. Instrument Aspek *Efficiency*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Execution Efficiency	Apakah performa aplikasi memerlukan banyak waktu?	✓	
5.	Execution Efficiency	Apakah proses pemanggilan data pada aplikasi memerlukan banyak waktu?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Operability	Apakah aktivitas program terlalu kompleks?	✓	
7.	Operability	Apakah alur penggunaan aplikasi terlalu rumit?	✓	
8.	Operability	Apakah navigasi pada aplikasi terlalu kompleks?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO RIZKY J

Pekerjaan : IT CONSULTANT

Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek Efficiency

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Execution Efficiency	Apakah performa aplikasi memerlukan banyak waktu?	✓	
5.	Execution Efficiency	Apakah proses pemanggilan data pada aplikasi memerlukan banyak waktu?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Operability	Apakah aktivitas program terlalu komplek?	✓	
7.	Operability	Apakah alur penggunaan aplikasi terlalu rumit?	✓	
8.	Operability	Apakah navigasi pada aplikasi terlalu komplek?	✓	

Lampiran 21. Hasil Uji *Integrity*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesapto Joeni Hanjoro
 Pekerjaan : Programmer
 Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL.

Tabel. Instrument Aspek *Integrity*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?		✓
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?		✓
7.	Instrumentation	Apakah aplikasi dikembangkan berdasarkan penerapan keilmuan?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
8.	Security	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?	✓	
9.	Security	Apakah " <i>admin</i> " dapat menambah " <i>Guru</i> " dan " <i>Siswa</i> " baru ?	✓	
10.	Security	Apakah " <i>admin</i> " dapat melakukan perubahan data pada " <i>Guru</i> " dan " <i>Siswa</i> " ?	✓	
11.	Security	Apakah " <i>admin</i> " dapat melihat nilai " <i>Guru</i> " dan " <i>Siswa</i> "?	✓	
12.	Security	Apakah " <i>Guru</i> " dapat melihat nilai " <i>Siswa</i> " ?	✓	
13.	Security	Apakah " <i>Siswa</i> " dapat melihat nilai siswa ?	✓	
14.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
15.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
16.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat melihat data?	✓	
17.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan perubahan data.	✓	
18.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan pengayaan nilai?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Anggyudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nokia E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Integrity*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?	✓	
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
7.	Instrumentation	Apakah aplikasi dikembangkan berdasarkan penerapan keilmuan?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
8.	Security	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?	✓	
9.	Security	Apakah " <i>admin</i> " dapat menambah " <i>Guru</i> " dan " <i>Siswa</i> " baru ?	✓	
10.	Security	Apakah " <i>admin</i> " dapat melakukan perubahan data pada " <i>Guru</i> " dan " <i>Siswa</i> " ?	✓	
11.	Security	Apakah " <i>admin</i> " dapat melihat nilai " <i>Guru</i> " dan " <i>Siswa</i> "?	✓	
12.	Security	Apakah " <i>Guru</i> " dapat melihat nilai " <i>Siswa</i> " ?	✓	
13.	Security	Apakah " <i>Siswa</i> " dapat melihat nilai siswa ?	✓	
14.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
15.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
16.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat melihat data?	✓	
17.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan perubahan data.	✓	
18.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan pengayaan nilai?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO RIZKY J
Pekerjaan : IT CONSULTANT
Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Integrity*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?	✓	
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
7.	Instrumentation	Apakah aplikasi dikembangkan berdasarkan penerapan keilmuan?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
8.	Security	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?	✓	
9.	Security	Apakah " <i>admin</i> " dapat menambah " <i>Guru</i> " dan " <i>Siswa</i> " baru ?	✓	
10.	Security	Apakah " <i>admin</i> " dapat melakukan perubahan data pada " <i>Guru</i> " dan " <i>Siswa</i> " ?	✓	
11.	Security	Apakah " <i>admin</i> " dapat melihat nilai " <i>Guru</i> " dan " <i>Siswa</i> "?	✓	
12.	Security	Apakah " <i>Guru</i> " dapat melihat nilai " <i>Siswa</i> " ?	✓	
13.	Security	Apakah " <i>Siswa</i> " dapat melihat nilai siswa ?	✓	
14.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
15.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat masuk aplikasi dan mengakses halaman <i>web</i> ?	✓	
16.	Security	Apakah " <i>Guru</i> " (tidak <i>login</i>) tidak dapat melihat data?	✓	
17.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan perubahan data.	✓	
18.	Security	Apakah " <i>Siswa</i> " (tidak <i>login</i>) tidak dapat melakukan pengayaan nilai?	✓	

Lampiran 22. Hasil Uji *Maintanability*

LEMBAR OBSEVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Editon, PHP dan MySQL

Nama : Socsapto Joeni Hantoro

Pekerjaan : Programmer

Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Maintanability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisennes	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisennes	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisennes	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
5.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
8.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?		✓
9.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Aryo Yudanta
Pekerjaan : Mobile - Developer
Perangkat Uji : Nokia E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*.

Tabel. Instrument Aspek Maintainability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Conciseness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Conciseness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Conciseness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
5.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
8.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
9.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO RIZKY J
Pekerjaan : IT CONSULTANT
Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek Maintainability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	
5.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
8.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
9.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
10.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
11.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
12.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
13.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
14.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

Lampiran 23. Hasil Uji *Flexibility*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesapto Jueni Hantoro
 Pekerjaan : Programmer
 Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Flexibility*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	
5.	Expandability	Dapatkah aplikasi dikembangkan dengan data yang lebih kompleks?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
7.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
13.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
14.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
15.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?		✓
16.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Aryyudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nokia E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek Flexibility

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	
5.	Expandability	Dapatkah aplikasi dikembangkan dengan data yang lebih komplek?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
7.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
13.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
14.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
15.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
16.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

LEMBAR OBSEVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO RIZKY
 Pekerjaan : IT CONSULTANT
 Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*.

Tabel. Instrument Aspek *Flexibility*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Concisenness	Apakah tiap fungsi dari aplikasi dirancang dengan jumlah baris kode yang sederhana?	✓	
2.	Concisenness	Apakah <i>library</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
3.	Concisenness	Apakah <i>framework</i> mendukung aplikasi untuk menjalankan fungsi dalam aplikasi?	✓	
4.	Consistency	Apakah konten soal pada aplikasi dirancang dengan bentuk yang sama?	✓	
5.	Expandability	Dapatkah aplikasi dikembangkan dengan data yang lebih kompleks?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
7.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
13.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
14.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
15.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
16.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

Lampiran 24. Hasil Uji *Testability*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesapto Joeni Hantoro
Pekerjaan : Programmer
Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL*.

Tabel. Instrument Aspek *Testability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?	✓	
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
7.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
8.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
9.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
10.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓
11.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
12.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
13.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
14.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
15.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Aryo yudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nolar E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Testability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?	✓	
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
7.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
8.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
9.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
10.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
11.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
12.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
13.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
14.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
15.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNARD RIZKY J

Pekerjaan : IT CONSULTANT

Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek Testability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Auditability	Apakah halaman <i>splash screen</i> berjalan dengan baik?	✓	
2.	Auditability	Apakah halaman <i>login</i> berjalan dengan baik?	✓	
3.	Auditability	Apakah halaman menu pilihan soal dapat berjalan dengan baik?	✓	
4.	Auditability	Apakah <i>radio button</i> pada pilihan jawaban berjalan dengan baik?	✓	
5.	Auditability	Apakah halaman upload nilai dapat berjalan dengan baik?	✓	
6.	Instrumentation	Apakah aplikasi dikembangkan sesuai alur pengembangan program?	✓	
7.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	

Lampiran 25. Hasil Uji *Reusability*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesanto Joeni Hantoro
 Pekerjaan : Programmer
 Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*.

Tabel. Instrument Aspek *Reusability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
2.	Hardware Independence	Apakah aplikasi Simulasi Tes dapat diinstal?	✓	
3.	Hardware Independence	Apakah <i>splash screen</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
4.	Hardware Independence	Apakah halaman <i>login</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
5.	Hardware Independence	Apakah menu pilihan soal dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Hardware Independence	Apakah halaman pilihan jawaban dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	
7.	Hardware Independence	Apakah <i>upload</i> nilai dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?		✓
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Software System Independence	Apakah aplikasi dikembangkan sesuai dengan bahasa pemrograman yang mendukung?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Argyudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Nokia E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Reusability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
2.	Hardware Independence	Apakah aplikasi Simulasi Tes dapat diinstal?	✓	
3.	Hardware Independence	Apakah <i>splash screen</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
4.	Hardware Independence	Apakah halaman <i>login</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
5.	Hardware Independence	Apakah menu pilihan soal dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
6.	Hardware Independence	Apakah halaman pilihan jawaban dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	
7.	Hardware Independence	Apakah <i>upload</i> nilai dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Software System Independence	Apakah aplikasi dikembangkan sesuai dengan bahasa pemrograman yang mendukung?	✓	

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERHADE RIZKY
Pekerjaan : IT CONSULTANT
Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Reusability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
2.	Hardware Independence	Apakah aplikasi Simulasi Tes dapat diinstal?	✓	
3.	Hardware Independence	Apakah <i>splash screen</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
4.	Hardware Independence	Apakah halaman <i>login</i> dapat berjalan pada <i>device</i> yang berbeda konfigurasi?	✓	
5.	Hardware Independence	Apakah menu pilihan soal dapat diakses pada <i>device</i> yang berbeda konfigurasi?		✓

No	Kategori	Pertanyaan	Ya	Tidak
6.	Hardware Independence	Apakah halaman pilihan jawaban dapat diakses pada <i>device</i> yang berbeda konfigurasi?		✓
7.	Hardware Independence	Apakah <i>upload</i> nilai dapat diakses pada <i>device</i> yang berbeda konfigurasi?	✓	
8.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
9.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	✓	
10.	Self Documentation	Apakah aplikasi dikembangkan dengan dokumentasi kode sumber yang terstruktur?	✓	
11.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
12.	Software System Independence	Apakah aplikasi dikembangkan sesuai dengan bahasa pemrograman yang mendukung?	✓	

Lampiran 26. Hasil Uji *Interoperability*

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Soesanto Joeni Hanibro
Pekerjaan : Programmer
Perangkat Uji : Handphone Nokia Asha 501

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP* dan *MySQL*.

Tabel. Instrument Aspek *Interoperability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Communication commonality	Apakah tampilan antarmuka aplikasi sesuai untuk standar pengembangan perangkat lunak?	✓	
2.	Communication commonality	Apakah aplikasi menggunakan protocol sebagai pengembangan?	✓	
3.	Communication commonality	Apakah aplikasi memerlukan <i>bandwith</i> untuk pengembangan?	✓	
4.	Data Commonality (Kesamaan Data)	Apakah komponen <i>database</i> dalam aplikasi sudah sesuai dalam penggunaanya?	✓	✓
5.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Bayu Aryoyudanta
Pekerjaan : Mobile Developer
Perangkat Uji : Noken E-63

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek *Interoperability*

No	Kategori	Pertanyaan	Ya	Tidak
1.	Communication commonality	Apakah tampilan antarmuka aplikasi sesuai untuk standar pengembangan perangkat lunak?	✓	
2.	Communication commonality	Apakah aplikasi menggunakan protocol sebagai pengembangan?	✓	
3.	Communication commonality	Apakah aplikasi memerlukan <i>bandwith</i> untuk pengembangan?	✓	
4.	Data Commonality (Kesamaan Data)	Apakah komponen <i>database</i> dalam aplikasi sudah sesuai dalam penggunaannya?	✓	
5.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓

LEMBAR OBSERVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : BERNADO Rizky S
Pekerjaan : IT CONSULTANT
Perangkat Uji :

Berilah tanda *checklist* (✓) pada kolom pilihan yang sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition*, *PHP* dan *MySQL*.

Tabel. Instrument Aspek Interoperability

No	Kategori	Pertanyaan	Ya	Tidak
1.	Communication commonality	Apakah tampilan antarmuka aplikasi sesuai untuk standar pengembangan perangkat lunak?	✓	
2.	Communication commonality	Apakah aplikasi menggunakan protocol sebagai pengembangan?	✓	
3.	Communication commonality	Apakah aplikasi memerlukan <i>bandwith</i> untuk pengembangan?	✓	
4.	Data Commonality (Kesamaan Data)	Apakah komponen <i>database</i> dalam aplikasi sudah sesuai dalam penggunaannya?	✓	
5.	Generality	Apakah aplikasi dapat dikembangkan dengan banyak fungsi?	✓	
6.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	

No	Kategori	Pertanyaan	Ya	Tidak
7.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Lampiran 27. Hasil Uji *Usability*

LEMBAR OBSEVASI PENGGUNA

Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition, PHP dan MySQL

Nama : Platmilah
Pekerjaan : Siswa SMN 3 Kk
Perangkat Uji : Nokia E63

Berilah tanda melingkar (O) pada kolom penilaian sesuai dengan pendapat anda selaku responden terhadap penggunaan Aplikasi Simulasi Tes Berbasis *Mobile Application* Menggunakan Bahasa Pemrograman *Java Micro Edition, PHP dan MySQL*.

Tabel. Computer System Usability Questionnaire J.R Lewis

No.	Pertanyaan	Skor
1.	Secara keseluruhan, saya merasa puas dengan kemudahan penggunaan sistem ini.	1 2 3 4 5 6 (7)
2.	Cara penggunaan sistem ini sangat simpel.	1 2 3 4 5 6 (7)
3.	Saya dapat menyelesaikan tugas saya dengan efektif ketika menggunakan sistem ini.	1 2 3 4 5 6 (7)
4.	Saya dapat dengan cepat menyelesaikan pekerjaan saya menggunakan sistem ini.	1 2 3 4 5 6 (7)
5.	Saya dapat menyelesaikan tugas saya dengan efisien ketika menggunakan sistem ini.	1 2 3 4 5 6 (7)
6.	Saya merasa nyaman menggunakan sistem ini.	1 2 3 4 5 6 (7)
7.	Sistem ini sangat mudah dipelajari.	1 2 3 4 5 6 (7)
8.	Saya yakin saya akan lebih produktif ketika menggunakan sistem ini.	1 2 3 4 5 6 (7)
9.	Jika terjadi error, sistem ini memberikan pesan pemberitahuan tentang langkah yang saya lakukan untuk mengatasi masalah.	1 2 3 4 5 6 (7)
10.	Kapanpun saya melakukan kesalahan, saya bisa kembali dan pulih dengan cepat.	1 2 3 4 5 6 (7)

11.	Informasi yang disediakan sistem ini sangat jelas.	1 2 3 4 5 6 7
12.	Mudah untuk menemukan informasi yang saya butuhkan.	1 2 3 4 5 6 7
13.	Informasi yang diberikan oleh sistem ini sangat mudah dipahami.	1 2 3 4 5 6 7
14.	Informasi yang diberikan sangat efektif dalam membantu menyelesaikan pekerjaan saya.	1 2 3 4 5 6 7
15.	Tata letak informasi yang terdapat di layar monitor sangat jelas.	1 2 3 4 5 6 7
16.	Tampilan sistem ini sangat memudahkan.	1 2 3 4 5 6 7
17.	Saya suka menggunakan tampilan sistem semacam ini.	1 2 3 4 5 6 7
18.	Sistem ini memberikan semua fungsi dan kapabilitas yang saya perlukan.	1 2 3 4 5 6 7
19.	Secara keseluruhan, saya sangat puas dengan kinerja sistem ini.	1 2 3 4 5 6 7

Yogyakarta, 30-10-2014

Responden,


R. Ramilah

No	Kategori	Pertanyaan	Ya	Tidak
8.	Self Documentation	Apakah kode program terdokumentasi untuk setiap fungsi pada aplikasi?	✓	
9.	Modularity	Apakah program dikembangkan dengan <i>library</i> atau <i>framework</i> ?	✓	
10.	Modularity	Apakah aplikasi Simulasi Tes dapat berjalan pada <i>emulator</i> dengan konfigurasi yang berbeda?		✓
11.	Simplicity	Apakah aplikasi mudah untuk di- <i>install</i> ?	✓	
12.	Simplicity	Apakah aplikasi Simulasi Tes mudah untuk dijalankan?	✓	
13.	Simplicity	Apakah tampilan sistem sangat memudahkan?	✓	
14.	Simplicity	Apakah informasi yang diberikan sangat mudah dipahami?	✓	
15.	Simplicity	Apakah soal pembelajaran dapat dipahami saat mengerjakan?		✓

Lampiran 28. Surat Keterangan Telah Penelitian

	PEMERINTAH KOTA YOGYAKARTA DINAS PENDIDIKAN SMK NEGERI 3 Jalan W. Monginsidi No. 2 Yogyakarta 55233 Telp./Fax. (0274) 513503 Website: www.smkn3jogja.sch.id Email: humas@smkn3jogja.sch.id	<table border="1" style="margin-left: auto;"><tr><td>F/62/TU/13</td></tr><tr><td>20 Agustus 2013</td></tr></table> <div style="margin-top: 10px;"><div style="font-size: small; margin-left: 5px;">Management System ISO 9001:2008 www.tuv.rwth-id 5105044805</div></div>	F/62/TU/13	20 Agustus 2013
F/62/TU/13				
20 Agustus 2013				

SURAT KETERANGAN TELAH MELAKSANAKAN PENELITIAN
Nomor : 070 / 1586

Yang bertanda tangan dibawah ini :

Nama	: Drs. Aruji Siswanto
NIP	: 19640507 199010 1 001
Jabatan	: Kepala Sekolah

Menerangkan bahwa :

Nama	: Rizam Yudinar
NIM	: 08502244021
Program Studi	: Pendidikan Teknik Elektronika
Universitas	: Universitas Negeri Yogyakarta
Fakultas	: Teknik

Bahwa yang bersangkutan telah melaksanakan Penelitian dengan judul : “Pengembangan Dan Analisis Aplikasi Simulasi Tes Berbasis Mobile Application Menggunakan Bahasa Pemrograman Java Micro Edition , PHP Dan Mysql ”

Demikian surat keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Yogyakarta ,20 Nopember 2014
Kepala Sekolah



Drs. Aruji Siswanto
NIP. 19640507 199010 1 001